NASA Technical Memorandum 86397

P- 46

IN - 03

65661

TRAFFIC SCENARIO GENERATION TECHNIQUE
FOR PILOTED SIMULATION STUDIES

David H. Williams and Douglas C. Wells

APRIL 1985

Date for general release ___April 30, 1987___

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

SUMMARY

Piloted simulation studies of cockpit traffic display concepts require the development of representative traffic scenarios. With the exception of specific aircraft interaction issues, most research questions can be addressed using traffic scenarios consisting of prerecorded aircraft movements merged together to form a desired traffic pattern. Prerecorded traffic scenarios have distinct research advantages, allowing control of traffic encounters with repeatability of scenarios between different test subjects. This paper describes a technique for generation of prerecorded jet transport traffic scenarios suitable for use in piloted simulation studies. Individual flight profiles for the aircraft in the scenario are created interactively with a computer program designed specifically for this purpose. The profiles are then time-correlated and merged into a complete scenario. This technique has been used to create traffic scenarios for the Denver, Colorado area with operations centered at Stapleton International Airport. Traffic scenarios for other areas may also be created using this technique, with appropriate modifications made to the navigation fix locations contained in the flight profile generation program.

INTRODUCTION

For the past several years, NASA Langley Research Center has been involved in a joint project with the Federal Aviation Administration (FAA) evaluating the Cockpit Display of Traffic Information (CDTI) concept. Langley's role in this project has been to study cockpit systems integration issues and pilot roles associated with using CDTI in the operation of jet transport aircraft. Operations in terminal areas, where traffic congestion is most severe and the potential benefits and hazards associated with traffic displays are most critical, have been the focus of the work at Langley. The primary research technique utilized for this effort has been piloted simulation.

A critical element in a piloted simulation study involving CDTI is the nature of the air traffic scenario utilized in the study. The complexity and realism required of a traffic scenario is a direct function of the research objectives of a particular study. Part-task experiments, such as those conducted at Langley involving in-trail self-spacing with CDTI (ref. 1 and 2), may be accomplished with a very minimal air traffic control environment. More elaborate experiments, involving full-systems simulation with the test crew subjected to a full workload cockpit environment, require a correspondingly more complete traffic scenario. This paper will describe a traffic scenario generation technique developed for such a full-systems experiment.

TRAFFIC ENVIRONMENT

The traffic environment modeled was an area within approximately a 100 n.mi. radius of the Denver Stapleton International Airport (see figure 1). Technically, all this airspace is the responsibility of the Denver Air Route Traffic Control Center (ARTCC). A specific area around the primary airport is re-delegated to the Denver Terminal Radar Approach Control (TRACON). The ARTCC's and TRACON's airspace is further divided, horizontally and vertically, into control sectors. Figure 1 illustrates the respective boundaries of the high and low altitude sectors of the Denver ARTCC. Included in the figure are the locations of several VORTAC (Very high frequency Omni-Range with an

associated TACtical air navigation transmitter) stations which reside in or near the simulated area. Figure 2 is a graphical representation of the horizontal boundaries of the Denver TRACON airspace when runway 26L is used for instrument approaches. The four metering fixes for Denver are shown in figure 2 (KIOWA, KEANN, DRAKO, and BYSON). The vertical delineation of the airspace is shown in figure 3, with the threshold of runway 26L as the zero reference in the figure. A table giving a summary of the major airspace segments and the significant events which occur in each segment is included in figure 3.

A typical sequence of events for a jet transport aircraft inbound to land at Denver Stapleton Airport would begin with air traffic control (ATC) issuing instructions for initial descent from cruise altitude when the aircraft is approximately 80 to 100 n.mi. from the airport. As the aircraft nears the lower vertical boundary of the particular ARTCC sector, coordination is initiated for a "handoff," that is, a transfer of control responsibility to the next controller. When coordination is completed, the pilot is instructed to tune his radio to the frequency of the accepting controller.

Once the aircraft enters the TRACON airspace (crossing one of the four metering fixes), it receives vector headings and altitude assignments which merge it into the sequence for the landing runway in use. A final handoff is made from the TRACON to the tower controller before the aircraft crosses the last navigation fix (outer marker) on the published approach procedure.

Departure procedures are typically the reverse. Before an aircraft begins to taxi, the pilot requests and receives a "clearance" which is an approval of, and/or changes to, the requested route of flight and altitude. This is followed by taxi instructions to the departure runway in use, takeoff clearance, and a transfer of control to the departure radar controller.

The departure radar controller, usually located in the TRACON, first establishes positive radar identification of the aircraft. He then issues vector headings and altitude assignments until the aircraft is clear of other known traffic and the pilot can assume his own navigation on course. As the aircraft nears the boundary of the TRACON's airspace, it is handed off to the adjoining facility.

Certain conditions were prestated for this particular study. Weather conditions were assumed which would require total reliance on the cockpit instruments for navigation and aircraft control. Based initially on conversations with Denver TRACON personnel, runway 26L was designated the landing runway and runways 35L and 35R for departures. Also, a maximum traffic density of 35 landings per hour was set. It was further assumed that all traffic in the scenarios would be jet transport aircraft.

After the test scenarios had been designated, traffic data were obtained from the Denver TRACON for the months of January through March of 1983. These representative data showed that during the sample period instrument weather conditions were present 9.9% of the time. During those periods of instrument weather, the runway 26L/35 arrival/departure combination was in use 52.6% of the time. Also, the mean arrival rate on runway 26L during the Instrument Flight Rules (IFR) periods was 35.9 aircraft per hour, with 89% of the aircraft being jet transport types.

CREATION OF TRAFFIC SCENARIO

The creation of a complete traffic scenario is a multistep, iterative process. First, the general characteristics of the scenario must be defined. Next, individual profiles of aircraft flying the desired approach and departure paths in the scenario are created using the computer program described in Appendix A. Finally, the individual profiles are time-correlated and merged together into the complete scenario. A detailed description of this process is given in the following sections.

Definition Phase

The first step in creating a traffic scenario is to define the general characteristics of the scenario. The major consideration in this definition is that the scenario can be designed to provide a specific traffic pattern for the test subjects to encounter while flying the simulator through the prerecorded scenario. The nature of this traffic pattern is dictated by the research objectives of the study which will be using the scenario. The desired pattern must conform to the traffic environment being simulated, which in this case is present-day Denver under instrument weather conditions. With these considerations in mind, the characteristics of the scenario can be defined.

The arrival or departure routing which the simulator will be flying must be chosen. The next step is to define the geometry of any specific encounters desired between the simulator and any particular traffic aircraft. This may be for an in-trail following case, a crossing situation or any desired encounter which could occur in the simulated traffic environment. The general routing of the encounter aircraft must then be defined from some appropriate initial conditions, through the encounter condition on to some final condition such as landing or leaving the terminal area. Finally, the flight paths of the remaining background traffic must be defined such that the density and sequencing of traffic is consistent with the simulated environment. The number of aircraft chosen for background traffic must be sufficient to fully encompass the simulator within the traffic pattern without having extraneous traffic which would burden the real-time simulation. This is not critical at this stage since unnecessary traffic will be eliminated during the merge phase.

Profile Generation Phase

Once the general routings of the simulator and the traffic aircraft have been determined, the individual flight profiles of each aircraft must be generated. A computer program written specifically for this purpose is described in Appendix A. This program provides for computation of the flight profile of a jet transport aircraft using interactive or predefined commands, which emulate present-day air traffic control techniques. The program was designed to enable rapid generation of aircraft flight profiles which are realistic for the simulated environment and, at the same time, can be controlled to exhibit the desired encounter geometry for the subject research study. In order to provide this flexibility, the program was restricted to modeling the flight characteristics of the jet transport aircraft without specifying any air traffic control strategy. All the commands necessary to generate a complete flight profile consistent with the chosen traffic

environment were provided. However, the operator of the program must determine the proper sequence of commands in order to create a realistic profile. Once a profile has been created, a time history of the pertinent state variables which define the profile can be saved on a data file for later use in the traffic scenario. In addition, a data file of commands used in creating the profile is generated. This command data file enables the user to recreate the same profile without the need to interactively input all the commands again. Fine tuning of the profile can also be accomplished by editing the command data file and then rerunning the flight profile program with the edited command file.

The profile generation phase requires an in-depth understanding of the air traffic control techniques employed within the simulated traffic environment. The first step in creating the profiles is, therefore, to review the techniques used by air traffic control for the desired situation. Ideally, the assistance of air traffic controllers from the subject facilities should be obtained. In addition, the Air Traffic Controller's Handbook and the Airman's Information Manual (ref. 3 and 4) provide rules and guidelines which should be observed.

Interactive generation of a profile using the computer program described in Appendix A is quite straightforward. The user will be prompted for initial conditions for the aircraft. Command options will then be provided allowing the user to specify new airspeed or Mach, altitude, heading, landing gear position, and ILS approach to runway 26L. Once the desired command options (one or more) are selected, the user specifies the segment distance for the aircraft to fly before new command options may be entered. At any point during the generation of a profile, when the command option menu is displayed, the user may review the profile by selecting a plotting option which graphically depicts the flight profile up to the current conditions. The user may then resume generation of the profile, save the profile data, or start over. Throughout this process, the commands input by the user are saved on a data file to enable rapid reconstruction of the profile without requiring interactive inputs. The actual profile data are only saved at user request following the completion of the profile generation. These data consist of time, altitude, groundspeed, airspeed, ground positions, heading, and bank angle. The data are in a format to be read by the scenario merge phase computer program discussed in the next section.

Merge Phase

Once the flight profiles of the individual aircraft have been created, they need to be merged together into a single complete scenario. This is done by time-correlating the flight profiles of the traffic aircraft with the profile representing the desired flight path of the simulator in order to produce the traffic geometry as specified in the scenario definition phase. Ground position time histories of the traffic aircraft are compared with that representing the simulator in order to determine a time bias for when the traffic profile should begin in order to arrive at the desired location at the desired time. The traffic profiles are adjusted by the time biases determined in this manner, and are then combined to form a time history of all traffic based on the scenario time of the baseline profile representing the simulator. In practice, it has been found convenient to use the final approach fix (outer marker) of the landing runway as the geographical position

for time-correlating the approach traffic. This provides a common point which all approach aircraft cross. The time when the simulator crosses the outer marker is defined as the baseline scenario time. The desired time for each approach aircraft can be referenced to this baseline in order to obtain the scenario time of arrival for the aircraft, which provides the desired approach spacing and traffic density (landings per hour). The time biases for the approach aircraft are then determined by subtracting the actual time they would cross the marker from the desired scenario time of arrival for that aircraft. Departure and/or conflict traffic must be time-correlated based on the desired geometry at some arbitrary geographical location in relation to the reference profile of the simulator. A computer program was developed to generate the merged scenario, given the individual profiles and time biases. A companion program provides plotting capability to view the traffic geometry of the merged scenario at any desired time in the scenario.

Sample Scenario

Figures 4a through 4d illustrate a traffic scenario created using this traffic generation technique. The figures represent "snapshots" of the horizontal traffic situation at various times in the scenario. Each figure is centered at the threshold of runway 26L in Denver with magnetic north being up and east to the right. The map features are the same as those described in the Interactive Plotting section of Appendix A. The aircraft representing the simulator which is to fly through the scenario (referred to as "reference aircraft") is depicted as a circular symbol as identified in figure 4a. The traffic aircraft are depicted as squares. The direction of travel for each aircraft in the scenario is given by straight line path predictor vectors extending from each aircraft symbol. The numbers adjacent to the predictor vectors of the traffic aircraft refer to the aircraft identification numbers in Table I. The initial condition of the scenario is shown in figure 4a, with the reference aircraft crossing the ELBEC intersection at cruise altitude enroute to Denver. Figure 4b shows the traffic pattern with the reference aircraft just entering the TRACON airspace, approximately 30 n.mi. from the Denver Airport. A potential conflict situation between the reference aircraft and a departing aircraft is shown in figure 4c. Finally, figure 4d shows the traffic situation when the reference aircraft crosses the outer marker on final approach to the runway.

Scenario Definition.- The sample scenario was developed for use in a CDTI simulation study in which the test subjects in the simulator would be given the task of performing in-trail self-spacing on another aircraft from cruise altitude all the way to landing. The traffic environment would be Denver under instrument weather conditions. In addition to the self-spacing task, another requirement for the scenario was that a different aircraft should pass as close to the simulator as possible without causing an actual loss of required IFR separation (3 n.mi. horizontal or 1000 ft vertical). This encounter should appear to the simulator crew on their CDTI display as a possible conflict. Additional background traffic would be needed to include all possible aircraft that would either be within range of the CDTI display in the simulator or within the same ATC sector receiving radio communications on the same frequency as the simulator. Given these requirements, the aircraft in the scenario could be defined.

The flight profile which the simulator would follow must allow for an extended period of in-trail following of another aircraft, as well as the opportunity for a close encounter with a third aircraft. Typical operations at Denver involve long lines of aircraft crossing one of four entry cornerposts into the terminal area. This situation is ideal for in-trail following from any direction into Denver. The choice for the actual routing of the simulator was, therefore, made to coincide with the best opportunity for an encounter with a third aircraft. At Denver, this situation exists when aircraft departing the terminal area must pass underneath the airspace of aircraft on the downwind leg of the approach to the landing runway. The most common runway configuration at Denver is landings on runway 26 and departures on runway 35. Given this situation, the most likely conflict will occur between approach traffic entering the terminal area from the southwest corner (BYSON on figure 2) turning downwind for runway 26L, and departure traffic from runway 35L turning 180 degrees left for a south departure (figure 4c). Initial conditions for the simulator were, therefore, chosen on a jet route southwest of Denver which would bring the aircraft through the BYSON cornerpost enroute to Denver. The actual starting point would be the ELBEC intersection (see figure 4a) where the simulator would be in cruise just prior to initiating its descent into Denver. These conditions would also be used for the lead aircraft which the simulator crew would be following with their CDTI. The encounter aircraft would be a southbound departure taking off from runway 35L.

The remainder of the traffic in the scenario were defined by sketching what a typical landing pattern would look like with the simulator and lead aircraft in several positions along their approach. Aircraft would be brought in from all approach corridors and merged into a single stream landing on runway 26L. Starting positions for aircraft approaching Denver from the three cornerposts other than BYSON were the holding fixes just outside of the cornerpost waypoints. An additional profile from the southwest was also defined which merged into the simulator's route prior to arriving at the holding fix for the BYSON approach. In addition to the south departures, an east departure was also defined which would be within range of the reference aircraft CDTI but would not be a conflict.

Profile generation.- Having defined the scenario, the individual profiles of the aircraft were generated using the program described in Appendix A. Seven individual profiles were generated. One profile, starting at ELBEC and ending at touchdown on runway 26L, would be used for both the simulator and lead aircraft which the simulator would be following. One profile each was created for the approach patterns arriving from the other three cornerposts. These, and one from the BACCA intersection merging at SHREW and crossing BYSON, were generated to represent the remainder of the arrival traffic. Two profiles were generated for the south and east bound departures. Air navigation charts, the Air Traffic Controllers Handbook, the Airman's Information Manual, and discussions with airline pilots and air traffic controllers from the Denver facilities were all utilized to insure that realistic airspeed, altitude, and heading commands were used in generating these profiles.

Merged scenario.- The individual profiles of the aircraft were time-correlated and merged into the scenario illustrated in figures 4a through 4d. Table I lists the individual aircraft in the scenario with their starting and ending positions, the location used for merging the profile, the desired and actual

times to this location, and the time bias needed to shift the profile to the desired merge time. For example, aircraft number four is the approach traffic which the simulator will be following to landing. The profile starts at ELBEC intersection and ends on runway 26L. The merge location chosen is the outer marker for runway 26L. The desired arrival time at this location is 1230 sec, which is 100 sec prior to when the simulator would arrive if it followed the exact profile listed as the reference aircraft. Since the actual time that aircraft four would arrive at the merge location is 1330 sec, a time bias of -100 sec must be applied to correct this arrival time to the desired time. In other words, the profile of aircraft four must be advanced 100 sec in order to sequence the aircraft into the traffic scenario. The desired arrival times for the remainder of the approach traffic were chosen to provide adequate spacing between consecutive aircraft and also result in the proper number of arrivals per hour as specified in the scenario definition.

REAL-TIME SIMULATION CONSIDERATIONS

The final, merged traffic scenario is utilized in the real-time simulation program as a sequential input data file which is read at regular time intervals during a simulation run. The input data file is created from the merged traffic scenario file, with conversions applied to match the simulation coordinate reference and time interval at which the real time program will read the input file. For example, the basic merged scenario data file is referenced to the runway 26L landing threshold with data at two second intervals. The DC-9 simulation program required navigation data referenced to the Denver VORTAC available at one second intervals. The simulation input data file was created by interpolating on the scenario data file to provide the necessary one-second update rate, with corrections applied to the data to transform the coordinate reference and match that used in the simulation program. The resulting input data file was then ready to be used by the simulation program without requiring conversions or interpolations while the real-time program was executing.

Two special provisions were incorporated in the DC-9 real-time simulation program to enhance the flexibility of the input traffic data file. First, a time bias specification for the traffic file was available while the simulation program was in a reset mode. This provision allowed advancing the traffic file by any desired time interval in order to adjust the initial positions of the traffic. Starting location for the simulator could then be chosen to be at any position along the designated approach path, with the traffic then adjusted to be at the proper position. This was especially useful for recreating and demonstrating particular traffic situations. In addition, the same traffic file could be used for both approach and departure scenarios simply by changing the initial time bias on the traffic. The second provision added to the simulation program specified a large table of flight identifiers to be assigned to the individual aircraft in the scenario. Individual simulation runs were assigned a particular offset value in this table used to correlate a flight identifier with an aircraft in the data file. This allowed the identical scenario to be used in different data runs with different flight identifiers on the traffic without changes to the traffic file or tables in the simulation program. This feature was most useful in masking an identical traffic scenario by merely changing the call signs of the aircraft involved.

The ATC communications with the aircraft in the traffic scenario were simulated by creating scripts for the controller to utilize during the "live" simulation runs. For each test condition, a traffic data file, time bias, and flight identifier offset were established. A printout of the traffic data file was used to determine the ATC communications which each aircraft would need to receive during the scenario. The time when the communication takes place, radio frequency it would occur on, and nature of the communication were then recorded. All communications to all aircraft were combined into one time-sequenced script for the controller to use during the simulation run. Any conflicts in communications were resolved by minor modifications to the individual flight profiles involved. During the actual simulation runs, the controller was provided with the running scenario time and radio frequency which the simulator cockpit had tuned. The controller would then issue or respond to any communication which would occur on the frequency tuned by the cockpit simulator. By using variations in his voice, the controller was able to mimic the responses of the traffic aircraft and the different ATC sector controllers during this series of CDTI experiments. An electronic voice modifier or several people reading scripts for the traffic could be used to provide more realism in the ATC partyline communications.

APPENDIX A
FLIGHT PROFILE COMPUTER PROGRAM DESCRIPTION

INTRODUCTION

This appendix describes the computer program used to create the individual flight profiles of the aircraft used in generating the traffic scenarios. The program was written in FORTRAN 5 for the CDC CYBER 175 computer system at NASA Langley Research Center. The program capabilities, including flight path command inputs, interactive plotting and data file generation are discussed in this appendix. A technical description of the equations of motion, mathematical models, and control logic used in the program is also provided.

PROGRAM CAPABILITIES

This program was written to provide a rapid means of creating realistic flight profiles of jet transport aircraft. It provides a set of command inputs allowing the user to specify the characteristics of the profile in a step-by-step process. The user may enter the command inputs interactively for each segment of the profile or from a taped data file. Interactive command inputs can be saved on a data file for recreating the profile. An interactive plotting capability is provided for viewing the horizontal and vertical trajectory parameters of the profile. When the user is satisfied with the profile, the trajectory parameters can be saved on a time history data file for use in the merged traffic scenario.

Trajectory Commands

A flight profile is created as a series of variable-distance segments, each defined by a list of trajectory commands. The segments are created in sequence with the user specifying the trajectory commands and distance to be flown during each successive flight segment. The available commands are airspeed, altitude, heading, landing gear, and ILS for 26L. The control logic of the program will "fly" the airplane model in such a manner to achieve the desired trajectory commands in a realistic fashion.

Airspeed commands are input as either indicated airspeed or Mach number. Altitude commands are entered as feet above mean sea level. Heading commands are entered as navigation headings or radar vectors. Navigation headings will be flown as a ground track with airplane heading corrected for wind. Radar vectors will be flown as airplane heading with ground track allowed to vary with the wind. Landing gear selection will toggle the gear up or down depending on current status. Selection of ILS for 26L will activate logic in the program which will capture and track the localizer and glideslope for landing on runway 26L in Denver. This option should not be selected until the airplane has been positioned for ILS intercept.

Trajectory commands may be input to the program interactively or through a data file. A new profile is typically generated using interactive inputs. The program will prompt the user for all necessary inputs during the interactive process. The steps involved in creating a profile in this manner are given below.

1. Initial conditions of the airplane are specified by the user.

2. Status of the airplane is displayed. User is prompted for new trajectory commands.

3. User inputs new trajectory commands and specifies a segment distance to be flown. Only those trajectory commands which change for the new segment need to be entered.

4. If a segment distance of zero is entered, the user will be given the options to plot the trajectory data up to this point on the screen, save the data, continue with the profile, start a new profile or exit the program. Continuing the profile returns the program to step 2, starting a new profile returns to step 1.

5. Program flies the airplane model for the specified segment distance. Control inputs are modified to achieve the trajectory commands specified in step 3. The program then returns to step 2.

The process outlined above provides the user with great flexibility in generating flight profiles. At the conclusion of any segment, the profile data can be reviewed graphically to insure that everything is proceeding properly. The profile generation may then continue, end, or start over at the discretion of the user. This allows for correction of errors in a profile prior to saving the data, and also permits creation of multiple profiles during one session.

The initial conditions and trajectory commands input to the program during the interactive creation of a profile are automatically written to a user-specified data file. This data file may then be used in a future session to recreate the profile without going through the interactive process. Also included in this data file are time and positions of the airplane at each segment of the profile. This feature was added to assist in creating radio communication scripts with air traffic control instructions for the airplanes in the scenarios. The times on the data file where the airplane was given new airspeed, altitude, or heading commands were correlated to the scenario times in order to determine when such instructions needed to be simulated.

Another feature of the command data files which proved useful was the ability to make minor adjustments in an existing profile. For example, extending the downwind leg of a given profile could be done by simply editing the command input file and increasing the appropriate segment distances. The profile could then be regenerated using the modified command input file. This greatly simplified the process of "fine tuning" a given traffic scenario to provide a specific traffic pattern.

Interactive Plotting

One of the more useful features of this program is the ability to obtain interactive plots of the horizontal and vertical trajectory parameters of the flight profile. The plotting was accomplished using the PLOT-10 library routines contained in the NASA Langley Computer System. A Tektronics graphics terminal, or another terminal equipped with PLOT-10 emulation software, is required to utilize this plotting capability. Hard copies of the plots could

be obtained from the screen or postprocessed using the plot vector files created by the PLOT-10 routines.

At the conclusion of any segment during the creation of a profile, the user can generate plots of the trajectory parameters. Figure A.1 illustrates an example of the plot of the horizontal trajectory of a flight profile. The plot shows the ground track of the airplane starting with the initial conditions and ending at the final position of the profile. Time marks are superimposed on the profile at user-specified intervals as indicated on the plot. These time marks are used to correlate the ground position of the airplane with time histories of the other trajectory parameters. The horizontal map is centered at the threshold of runway 26L in Denver, magnetic north being up, with range rings at 10 n.mi. intervals and symbols indicating important waypoints and intersections in the Denver area. Also illustrated on the map is a dashed line showing the extended centerline for runway 26L. The range scale for the plot is specified by the user in terms of n.mi. radius from Denver to be displayed. This arrangement is similar to the radar display used by air traffic controllers in Denver, and has been found to be most useful for providing proper heading vectors to the simulated aircraft during the generation of the flight profiles.

Time history plots are also available for altitude, indicated airspeed, and groundspeed of the profile. Figure A.2 illustrates the altitude and airspeed time histories of the profile shown in figure A.1. Altitude or airspeed at any point in the profile can be determined by cross-referencing the time marks on the horizontal plot with the corresponding time on the plots in figure A.2.

Data File Generation

The end result of this program is a data file containing the trajectory parameters of a specific flight profile. The parameters saved include time, altitude, groundspeed, indicated airspeed, x position (north) from runway 26L threshold, y position (east) from runway 26L threshold, and magnetic heading. The data are saved at 2 sec intervals throughout the entire profile. Since a complete approach profile can be up to 30 minutes in length, this can represent a large amount of data.

TECHNICAL DESCRIPTION

The purpose of this program is to generate a time history of the state variables describing the flight profile of a jet transport aircraft. This is accomplished by integrating simplified trajectory equations of motion over various segment lengths along the desired profile. The segments are user-defined of variable length, separated by command inputs which specify desired characteristics of the profile. These characteristics consist of the airspeed, altitude, and heading which the aircraft is to fly. Control logic within the program will modify the control variables in the equations of motion in order to achieve the specified flight conditions.

Equations of Motion

The trajectory equations describing the motion of a point mass over a flat earth were used to represent the airplane. These equations ignore all

short-term dynamics of the airplane and reduce the math model to a simple drag polar and engine thrust representation. These equations are given below:

$$m\dot{V} = T\cos\alpha - D - W\sin\gamma$$

$$(mV\cos\gamma)\,\dot{\Psi} = (L + T\sin\alpha)\sin\phi$$

$$mV\dot{\gamma} = (L + T\sin\alpha)\cos\phi - mg\cos\gamma \qquad\qquad \text{A.1}$$

$$\dot{h} = V\sin\gamma$$

$$\dot{x} = V\cos\gamma\cos\Psi$$

$$\dot{y} = V\cos\gamma\sin\Psi$$

The state variables describing the flight profile are:

$$V,\ \Psi,\ \gamma,\ h,\ x,\ y$$

The control variables are:

$$T,\ \phi,\ \alpha$$

where L and D are functions of α .

Mathematical Models

Aerodynamics.- The use of trajectory equations of motion reduces the mathematical description of airplane aerodynamics to simple lift and drag relationships. The equations used for these terms are:

$$L = C_L qs$$
$$\qquad\qquad\qquad\qquad \text{A.2}$$
$$D = C_D qs$$

where:

$$C_L = C_{L_{\alpha_0}}(f) + C_{L_\alpha}(f)\,\alpha$$

$$C_D = C_{D_0}(f) + C_{D_{C_L}}(f)\,C_L^2 + C_{Dgear}(f)$$

q = dynamic pressure

s = wing area

f = flap setting

$C_{L_{\alpha_0}}$ = lift coefficient at zero angle of attack

C_{L_α} = slope of lift curve

C_{D_0} = drag coefficient at zero lift

$C_{D_{C_L}}$ = slope of drag coefficient versus square of lift coefficient

Lift and drag coefficients are solely a function of angle of attack, flap setting, and landing gear position. Angle of attack is a control variable used by the control logic. Flap setting was programmed as a function of indicated airspeed. Landing gear selection was implemented as a main command input, and could be raised or lowered as desired by the operator.

Engine.- The only engine parameter needed for the equations of motion is thrust. Aircraft weight was assumed constant, thus eliminating the need for engine fuel flow. Engine dynamics were neglected in this model. Therefore, thrust was allowed to instantaneously vary between maximum climb thrust and idle thrust. The engine model consisted of tabulated engine thrust for max climb and idle power as a function of altitude and Mach number. The data for this model were obtained from reference 5 for a JT8D-7 turbofan engine operating under standard day conditions. Linear interpolation was used to obtain the maximum and minimum thrust available at a given altitude and speed.

Control Logic

The control variables available in the equations of motion are thrust, angle of attack, and bank angle. Vertical control of altitude and airspeed is accomplished by varying thrust and angle of attack while horizontal control utilizes bank angle. The control logic implemented in this program was designed to provide for changes in flight conditions in a manner consistent with typical operation of a jet transport. No attempt was made to model pilot response or autopilot logic which might be used to actually fly the aircraft through the specified flight conditions.

Vertical logic.- In order to capture and maintain specified altitude and airspeed conditions, four unique vertical flight phases are considered. These phases are level flight, climb, descent, and glideslope tracking. The control logic examines the altitude error between actual altitude and desired altitude in order to determine which flight phase is in effect. The glideslope tracking phase is a unique case which is identified by flags passed from the ILS tracking routine. Within each flight phase, thrust is set and a desired trim flight path angle and trim angle of attack are determined. The commanded angle of attack is then calculated using the following equation:

$$\alpha_c = \alpha_t + k \, (\gamma_o - \gamma_d) \qquad\qquad A3$$

where, α_c = commanded angle of attack

α_t = trim angle of attack for desired conditions

γ_o = previous flight path angle

γ_d = desired flight path angle

k = gain on gamma error (-1.5 used)

The desired flight path and thrust settings for the different flight phases are given below:

Level: $\gamma_d = 0$

T = thrust required for trim at desired speed and γ

If more than 3 knots too slow, $T = T_{max}$

If more than 3 knots too fast, $T = T_{min}$

Climb: $T = T_{max}$

γ_d = trim γ at T_{max} and desired speed

If more than 2 knots too slow, $\gamma_d = 0$.

Descent: $T = T_{min}$

γ_d = trim γ at T_{min} and desired speed

If more than 2 knots too fast, $\gamma_d = 0$.

Glideslope: $\gamma_d = -3$ degrees

T = thrust required for trim at desired speed and γ

If more than 3 knots too slow, $T = T_{max}$

If more than 3 knots too fast, $T = T_{min}$

14

Horizontal logic.- Turns are accomplished by adding a bank angle to compensate for heading errors. The following equations are used:

$$\Psi_{err} = \Psi_c - \Psi$$

$$\phi_{max} = 5. \, |\Psi_{err}|, \text{ limited to 25 degrees}$$

where

Ψ_{err} = heading error

Ψ_c = commanded heading

Ψ = current heading

ϕ_{max} = max allowable bank angle

The actual bank angle is changed at a rate of 2 degrees per second whenever a heading error exists. Bank angle is never permitted to exceed the maxiumum allowable as calculated in the above equation. The sign logic is designed such that the aircraft will always turn toward the commanded heading. As a result, it is not possible to command the aircraft to turn more than 180 degrees in either direction with a single commanded heading. If this situation is necessary, the turn must be commanded as a series of less than 180 degree heading changes in the desired direction.

PROGRAM LISTING

The following pages contain the program listing for the flight profile generation computer program. Although it is written in CDC FORTRAN 5, it can be easily modified for use on other computers which support FORTRAN 77 compilers. The plotting routine utilizes Tektronics Plot-10 subroutines which must be linked with the program in order to utilize the plotting capability.

```
      PROGRAM FLTPRO5(INPUT,OUTPUT,TAPE10,TAPE11,TAPE12,TAPE13,
     * TAPE14,TAPE15,TAPE16,TAPE17,TAPE18,TAPE19,TAPE20,TAPE21,
     * TAPE22,TAPE23,TAPE24,TAPE25,TAPE26,TAPE27,TAPE28,TAPE29)
C
C ****************************************************
C
C PROGRAM FLTPRO5 : DATE 08-10-83
C
C FTN5 VERSION
C
C ACCEPTS INPUT INTERACTIVELY OR VIA DATA FILE
C
C    INTERACTIVE : CREATES FILE OF USER INPUTS
C                  FOR USE AS DATA FILE IN
C                  SUBSEQUENT SESSION.
C
C    DATA FILE : READS USER-SPECIFIED FILE.
C                WRITES NEW COMMAND FILE ON SEPARATE TAPE.
C
C    TAPE10 THRU TAPE19 ARE AVAILABLE
C
C OUTPUT FLIGHT PROFILE DATA CAN BE SAVED ON
C TAPE20 THRU TAPE29.
C
C A MAXIMUM OF 5 PROFILES CAN BE CREATED AND
C SAVED DURING ONE SESSION.
C
C ****************************************************
C
C GENERATES APPROACH AND DEPARTURE FLIGHT
C PROFILES FOR A DC-9 CLASS AIRCRAFT.
C
C INCORPORATES LINEAR WIND MODEL
C
C    VW=VWO+(HP-5330.)*.0012
C
C WHERE: VWO=RUNWAY WIND
C        PSIW=WIND DIRECTION
C
C        ARE USER INPUTS
C
C ****************************************************
C
      COMMON/TB/TBCLA(8),TBCLAO(8),TBCLS(8),TBCDO(8),TBCDCL(8),
     *TBCDS(8),TBCDG(8),TBFLAP(8),XS(1000),YS(1000),
     *TIME(1000),VSAV(1000),HPSAV(1000),PSISAV(1000),
     *VCSAV(1000)
      DIMENSION ENHP(10),ENXMN(5),TTBL(10,5,2),FLAPL(8),
     *FLAPH(8)
C
C        ** DC-9 DATA **
C
      DATA TBFLAP/0.,1.,5.,15.,25.,50.,50.,50./
      DATA FLAPL/190.,165.,165.,145.,135.,100.,100.,100./
      DATA FLAPH/500.,200.,180.,175.,145.,135.,135.,135./
```

```
        DATA TBCLA/.096,.1,.113,.126,.136,.12,.12,.12/
        DATA TBCLAO/.04,.06,.19,.38,.5,1.32,1.32,1.32/
        DATA TBCLS/0.,0.,0.,0.,0.,0.,0.,0./
        DATA TBCDO/.021,.0305,.0343,.045,.0638,.141,.141,.141/
        DATA TBCDCL/.0484,.045,.0423,.0416,.0409,.0395,.0395,.0395/
        DATA TBCDG/.0232,.0232,.0219,.0194,.0168,.0105,.0105,.0105/
        DATA TBCDS/0.,0.,0.,0.,0.,0.,0.,0./
        DATA ENHP/0.,10000.,20000.,30000.,0.,0.,0.,0.,0.,0./
        DATA ENXMN/0.0,.2,.4,.6,.8/
        DATA (((TTBL(I,J,K),K=1,2),J=1,5),I=1,4)/
       *25200.,1700.,22400.,600.,20500.,-200.,19800.,-500.,
       *19600.,-1400.,
       *19800.,1000.,17300.,500.,16000.,-100.,15700.,-300.,
       *16000.,-900.,
       *14500.,800.,12900.,300.,12200.,0.,12100.,-400.,
       *12400.,-600.,
       *9600.,500.,9100.,500.,8800.,200.,8850.,0.,
       *9300.,-200./
        S=1000.
C
C
C
C ******** INITIALIZE VARIABLES *********
C
      1 ISPOIL=0
        ITRACE=0
        PHI=0.
        ILS=0
        ILOC=0
        IGLIDE=0
        TIME(1)=0.
        GAMTOL = .001
        AGAIN = -1.5
        HPIDLE = 15000.
        ROCALT = -3000.
C
C ******** INITIAL CONDITIONS **********
C
        PRINT 200
  200 FORMAT(//1X,'FLIGHT PROFILE PROGRAM--DC9',
       */,1X,'INTERACTIVE = 0, TAPED = 1')
        READ *, IMODE
C
        IF ( IMODE .EQ. 1 ) THEN
C
C<<<< READ INPUT DATA FROM TAPE 2 >>>>>>>>>
C
          PRINT 601
  601     FORMAT (1X,'INPUT TAPE NUMBER (10 THRU 19)')
          READ *, ITAPE
          PRINT 602
  602     FORMAT(1X,' OUTPUT TAPE (CANNOT BE SAME AS INPUT)')
          READ *,ITAPEW
          REWIND ITAPEW
          REWIND ITAPE
```

```
C
        READ (ITAPE,600) TITLE
  600   FORMAT ( A10 )
        READ (ITAPE,*) X , Y
        READ (ITAPE,*) WT, HP, VC, PSIM
        READ (ITAPE,*) INAV
        READ (ITAPE,*) VWO, PSIW
        READ (ITAPE,*) FLAP
        READ (ITAPE,*) IGEAR
        READ (ITAPE,*) ITRIM
        IF (ITRIM .EQ. 1 ) THEN
          READ (ITAPE,*) ROC
          ITHR = 2
        ELSE
          READ (ITAPE,*) ITHR
        ENDIF
      ELSE
C
C<<<<<< INPUT DATA INTERACTIVELY >>>>>>>>
C
        PRINT 203
  203   FORMAT (1X,'TAPE NUMBER FOR COMMAND DATA OUTPUT',
     *            /,' 10 THRU 19 AVAILABLE')
        READ *, ITAPEW
        REWIND ITAPEW
        ITAPE=ITAPEW
C
        PRINT 201
  201   FORMAT (1X,'INPUT TITLE (10 CHARS MAX)')
        READ (*,600) TITLE
        PRINT 202
  202   FORMAT (1X,'INPUT X (NORTH), Y (EAST)')
        READ *,X,Y
        PRINT 100
  100   FORMAT(1X,'WT,ALT,KCAS,HEADING')
        READ *,WT,HP,VC,PSIM
C
        PRINT 120
  120   FORMAT(1X,'1=NAV HEADING, 0=VECTOR')
        READ *,INAV
        PRINT 121
  121   FORMAT(1X,'WIND SPEED,DIRECTION')
        READ *,VWO,PSIW
   17   PRINT 101
  101   FORMAT(1X,'FLAP SETTING')
        READ *,FLAP
        PRINT 102
  102   FORMAT(1X,'GEAR,   0=UP,1=DN')
        READ *,IGEAR
        PRINT 103
  103   FORMAT(1X,'TRIM CONDITION, 0=THRUST,1=ROC')
        READ *,ITRIM
        IF (ITRIM .EQ. 1) THEN
C
            PRINT 104
```

```
  104      FORMAT(1X,'CLIMB RATE (FPM)')
           READ *,ROC
           ITHR=2
C
         ELSE
C
           PRINT 105
  105      FORMAT(1X,'THRUST (0=IDLE, 1=CLIMB)')
           READ *,ITHR
C
         END IF
C
       ENDIF
C
C<<< WRITE COMMANDS ON OUTPUT FILE >>>
C
         WRITE (ITAPEW,600) TITLE
         WRITE (ITAPEW,*) X, Y, ' X-NORTH, Y-EAST'
         WRITE (ITAPEW,*) WT,HP,VC,PSIM,'  WT,ALT,KCAS,HEADING'
         WRITE (ITAPEW,*) INAV, '  INAV'
         WRITE (ITAPEW,*) VWO, PSIW,' WIND SPEED, DIR'
         WRITE (ITAPEW,*) FLAP,'  FLAP SETTING'
         WRITE (ITAPEW,*) IGEAR,'  GEAR POS (0 = DN,1 = UP)'
         WRITE (ITAPEW,*) ITRIM,'  ITRIM ( 0 = THRUST, 1 = ROC)'
         IF ( ITRIM .EQ. 1 ) THEN
           WRITE (ITAPEW,*) ROC,' CLIMB RATE (FPM)'
         ELSE
           WRITE (ITAPEW,*) ITHR,'  THRUST ( 0 = IDLE, 1 = CLIMB)'
         END IF
C
C<<< INITIALIZE VARIABLES BASED ON INPUTS >>>
C
       X = X * 6076.
       Y = Y * 6076.
       IF ( INAV .EQ. 1  .OR.  VWO .EQ. 0. ) PSI = PSIM
       PSIGC = PSI
       VCOM = VC
       HPCOM = HP
       PSICOM = PSIM
C
       DO 15 I = 8,1,-1
       IF ( FLAP .EQ. TBFLAP(I)) GO TO 16
   15 CONTINUE
   16 IFLAP = I
       CALL FLAPS (FLAPL,FLAPH,VC,IFLAP)
C
       THETA=(1-.00000687*HP)
       DEL=THETA**5.256
       XMN=5*(((( 1.+.2*(VC/661.5)**2)**3.5-1.)/DEL+1.)**.286-1.)
       XMN=SQRT(XMN)
       OS=1481.*DEL*XMN*XMN*S
       VT=29.04*SQRT(518.7*THETA)*XMN
       IF(VWO.EQ.0.)VG=VT
       CALL WIND(HP,VWO,VT,VG,PSI,PSIW,PSIM,INAV)
       PRINT 90,VT,VG,PSIM,PSI
```

```fortran
   90 FORMAT(1X,'VT=',F10.2,' VG=',F10.2,',PSIM=',F10.2,' PSI=',
     *F10.2)
        IF(INAV.EQ.1)PSICOM=PSIM
        CALL ENG(HP,XMN,ENHP,ENXMN,TTBL,TMAX,TMIN)
        IF(ITHR.EQ.0)T=TMIN
        IF(ITHR.EQ.1)T=TMAX
C
C        ** INITIAL AERO TERMS **
C
        CLAO=TBCLAO(IFLAP)
        CLA=TBCLA(IFLAP)
        CDO=TBCDO(IFLAP)
        IF(IGEAR.EQ.1)CDO=CDO+TBCDG(IFLAP)
        CDCL=TBCDCL(IFLAP)
C
C        ** INITIAL TRIM **
C
        IF(ITRIM.EQ.1)GAMMA=ASIN((ROC/(VT*101.28)))
        CALL TRIM(ITRIM,T,WT,QS,CLAO,CLA,CDO,CDCL,GAMMA,ALPHA,
     * PHI)
        ROC=VT*SIN(GAMMA)*101.28
        GAMMA=GAMMA*57.3*VT/VG
C
C
C        ** INITIALIZE MAIN LOOP **
C
        ICNT=2
        XS(2)=X/6076.
        YS(2)=Y/6076.
        VSAV(2)=VG
        VCSAV(2)=VC
        HPSAV(2)=HP
        PSISAV(2)=PSIM
        TIME(2)=0.
        RANGE = 0.
C
C********** MAIN LOOP ************
C
    2 CFLAP=TBFLAP(IFLAP)
C
        IF ( IMODE .EQ. 1 ) THEN
C
C<<<<<<< READ TIME HEADER WITH DUMMY ARGUMENT >>>>>>
C
        READ (ITAPE,600) DUMM
        READ (ITAPE,600) DUMM
        READ (ITAPE,600) DUMM
        END IF
C
C<<<<<<< WRITE TIME HEADER >>>>>>>
C
        WRITE (ITAPEW,*) ' '
        WRITE (ITAPEW,*) ' *** TIME = ',TIME(ICNT), ' ***'
     *    ,'     RANGE = ',RANGE
        WRITE (ITAPEW,*) ' NORTH = ', XS(ICNT), '   EAST = ', YS(ICNT)
```

```
C
      CALL COMMAND(HP,VC,PSIM,HPCOM,VCOM,PSICOM,SDIST,IGEAR,
     *CFLAP,ITRACE,ILS,INAV,VT,VWO,PSIW,PSIGC,IMODE,ITAPE,ITAPEW)
      IF(SDIST.EQ.O.)GO TO 5
      DCHECK=0.
    4 ISAVE=0
    3 CALL ENG(HP,XMN,ENHP,ENXMN,TTBL,TMAX,TMIN)
C
      IF (ILS .EQ. 1) THEN
C
        CALL ILS26L(VT,VG,X,Y,HP,HPCOM,PSI,PSIW,PSICOM,ILOC,
     *   IGLIDE,VWO)
C
      END IF
C
C
      IF (VCOM .LT. 1.) THEN
C
        VCCOM =
     $   1481.*SQRT((DEL*((VCOM**2/5.+1.)**3.5-1.)+1.)**.286-1.)
        XMNTRM=VCOM
C
      ELSE
C
        VCCOM=VCOM
        XMNTRM =
     $   SQRT(5*(((((1.+.2*(VCCOM/661.5)**2)**3.5-1.)/DEL+1.)**.286-1.))
C
      END IF
C
      QSTRIM=1481.*DEL*XMNTRM**2*S
C
      DH = HP - HPCOM
      DV = VC - VCCOM
      CLAO = TBCLAO(IFLAP)
      CLA = TBCLA(IFLAP)
      CDO = TBCDO(IFLAP)
      IF (IGEAR .EQ. 1) CDO = CDO + TBCDG(IFLAP)
      CDCL = TBCDCL(IFLAP)
C
C-------------- MAIN CONTROL LOGIC ------------------
C
      IF ( ILS .EQ. 1 .AND. IGLIDE .EQ. 1 ) THEN
C
C        ON GLIDESLOPE
C
        ITRIM = 1
        DGAM = -3./57.3 * VG/VT
        CALL TRIM(1,T,WT,QSTRIM,CLAO,CLA,CDO,CDCL,DGAM,DALP,PHI)
        IF (DV .LT. -3. ) T=TMAX
        IF (DV .GT. 3.  ) T=TMIN
C
      ELSE IF ( DH .GT. 50. ) THEN
C
C        DESCENT
```

```
C
          ITRIM = 0
          T = TMIN
          IF ( HP .GT. HPIDLE .AND. ROC .LT. ROCALT ) THEN
            ITRIM = 1
            DGAM = ASIN ( ROCALT / (VT * 101.28) )
C
          ELSE IF ( DV .GE. 2. ) THEN
C
C            LEVEL OFF IF TOO FAST
C
            ITRIM = 1
            DGAM = 0.
C
C          END DESCENT
C
          END IF
C
       ELSE IF ( DH .LT. -50. ) THEN
C
C          CLIMB
C
          T = TMAX
          ITRIM = 0
          IF ( DV .LT. -2. ) THEN
C
C            LEVEL OFF IF TOO SLOW
C
            DGAM = 0.
            ITRIM = 1
C
C          END CLIMB
C
          END IF
C
       ELSE
C
C          LEVEL
C
          ITRIM = 1
          DGAM = -DH * .001
          CALL TRIM(1,T,WT,QSTRIM,CLAO,CLA,CDO,CDCL,DGAM,DALP,PHI)
          IF (DV .LT. -3. ) T = TMAX
          IF (DV .GT.  3. ) T = TMIN
C
       END IF
C
C-------------------------------------------------------------------
C
    75 CONTINUE
C
C -- FIND TRIM ALPHA AND GAMMA --
C
       PHI = PHI/57.3
    77 XCL = (WT*COS(DGAM)/COS(PHI))/QSTRIM
```

```fortran
      DALP = (XCL-CLAO)/CLA
      IF(ITRIM .EQ. 1)GO TO 76
      XCD = CDO + CDCL*XCL*XCL
      XDD = XCD*QSTRIM
      DGAM1 = ASIN((T*COS(DALP/57.3)-XDD)/WT)
      IF( ABS(DGAM-DGAM1) .LT. GAMTOL ) GO TO 76
      DGAM = DGAM1
      GO TO 77
   76 DGAM = DGAM*57.3
      PHI = PHI*57.3
C
C --- MODIFY ALPHA BASED ON GAMMA ERROR ---
C
      ALPHA = DALP + AGAIN*(GAMMA-DGAM)
C
      IF(ALPHA .GT. 12.)ALPHA=12.
      IF(ALPHA .LT. 0.)ALPHA=0.
      IF ( INAV .NE. 1 ) GO TO 85
      CALL WIND(HP,VWO,VT,VGDUM,PSIGC,PSIW,PSICOM,1)
   85 CALL BANK(PSIM,PSICOM,PHI)
      CALL AERO(IFLAP,IGEAR,ISPOIL,ALPHA,CL,CD)
      XL=CL*QS
      D=CD*QS
      CALL SOLVE(WT,ALPHA,GAMMA,PSI,PHI,T,XL,D,X,Y,HP,DR,ROC,
     *VG,VT,PSIM)
      CALL ATMOS(HP,VT,VC,XMN,QS,DEL)
      CALL WIND(HP,VWO,VT,VG,PSI,PSIW,PSIM,INAV)
      QS=QS*S
      CALL FLAPS(FLAPL,FLAPH,VC,IFLAP)
      IF(IFLAP.EQ.0)GO TO 32
      ISAVE=ISAVE+1
      DCHECK=DCHECK+DR
      RANGE = RANGE + DR
      IF(ISAVE.LT.10)GO TO 3
      ICNT=ICNT+1
      TIME(ICNT)=TIME(ICNT-1)+2.
      IF(ITRACE.NE.1)GO TO 87
      PRINT 91,TIME(ICNT)
      PRINT 91,VC,HP,ALPHA,T
      PRINT 91,PSI,PHI,ROC
   91 FORMAT(1X,'TRACE',4F8.2)
   87 CONTINUE
      XS(ICNT)=X/6076.
      YS(ICNT)=Y/6076.
      VSAV(ICNT)=VG
      VCSAV(ICNT)=VC
      HPSAV(ICNT)=HP
      PSISAV(ICNT)=PSIM
      IF(DCHECK.LT.SDIST)GO TO 4
      GO TO 2
    5 CONTINUE
C
      PRINT 301
      READ *, IPLOTF
      IF ( IPLOTF .EQ. 1 ) CALL PLOT (ICNT)
```

```fortran
C
      PRINT 300
      READ *, ISAV
      IF (ISAV .EQ. 1) THEN
C
C<<<<<<<<<<< SAVE FLIGHT PROFILE DATA >>>>>>>>>>>
C
         IOUT = ITAPE + 10
         WRITE (IOUT,600) TITLE
         DO 25 I = 2,ICNT
         WRITE (IOUT,*) TIME(I), HPSAV(I), VSAV(I), PSISAV(I),
     *      XS(I), YS(I), VCSAV(I)
   25    CONTINUE
         PRINT 401, IOUT
      ENDIF
C
      PRINT 400
      READ *, ICONT
      IF ( ICONT .EQ. 1 ) GO TO 2
  300 FORMAT(1X,'SAVE DATA (0=NO,1=YES)')
  301 FORMAT(1X,'PLOT DATA (1=YES)')
  400 FORMAT(1X,'CONTINUE (1=YES)')
  401 FORMAT(1X,'DATA SAVED ON TAPE',I2)
   32 PRINT 31
   31 FORMAT(1X,'FLAP ERROR')
   30 PRINT 500
  500 FORMAT(1X,'NEW CASE (1=YES)')
      READ *,INEW
      IF(INEW.EQ.1)GO TO 1
      END
      SUBROUTINE TRIM(ITRIM,T,WT,QS,CLAO,CLA,CDO,CDCL,
     *GAM,ALPHA,PHI)
C
C ****************************************************
C
C     THIS ROUTINE CALCULATES TRIM ANGLE OF ATTACK
C     FOR :
C
C        ITRIM = 0 ; FIXED THRUST, VARIABLE GAMMA
C
C        ITRIM = 1 ; FIXED GAMMA, VARIABLE THRUST
C
C ****************************************************
C
      ACAL=0.
C
      IF (ITRIM .EQ. 1) THEN
        TINIT = 1000.
C
      ELSE
        TINIT = T
C
      END IF
C
   11 AINIT=ACAL
```

```fortran
   20 XL=WT-TINIT*AINIT/57.3
      CL=XL/QS
      CL=CL/COS(PHI/57.3)
      ACAL=(CL-CLA0)/CLA
      ATEST=ABS(AINIT-ACAL)
      IF(ATEST.GT.0.05)GO TO 11
      CD=CD0+CDCL*CL*CL
      D=CD*QS
      IF(ITRIM.EQ.0)GO TO 30
      TC=D+WT*GAM
      TEST=ABS(TINIT-TC)/TINIT
      IF(TEST.LT.0.005)GO TO 21
      TINIT=TC
      GO TO 11
   21 T=TC
      GO TO 40
   30 GAM=(T-D)/WT
   40 ALPHA=ACAL
      RETURN
      END
      SUBROUTINE COMMAND(HP,VC,PSI,HPCOM,VCCOM,HCOM,SDIST,IGEAR,
     *CFLAP,ITRACE,ILS,INAV,VT,VWO,PSIW,PSIGC,IMODE,ITAPE,ITAPEW)
C
C ******************************************************
C
C THIS ROUTINE ACCEPTS INPUT TO ALTER THE
C FOLLOWING COMMAND PARAMETERS:
C
C     HPCOM (DESIRED ALTITUDE)
C     VCCOM (DESIRED AIRSPEED)
C     HCOM (DESIRED HEADING)
C     TRACE (DEBUG TRACE PRINT OUT)
C     ILS (ILS MODE SELECT)
C     IGEAR (LANDING GEAR SELECT)
C
C ******************************************************
C
      DIMENSION ICOM(6)
      PRINT 10,HP,VC,PSI
   10 FORMAT(/,1X,'ALT=',F8.0,/,1X,'KCAS=',F7.0,/,1X,'PSI=',F8.1)
      PRINT 200,CFLAP
  200 FORMAT(1X,',FLAPS ',F3.0)
   11 CONTINUE
C
      IF (IMODE .EQ. 0 )
     *  PRINT 20
   20 FORMAT(/,1X,'NEW COMMAND',/,1X,'1=KCAS,2=ALT,3=PSI',/,
     *1X,'4=TRACE,5=ILS26L,6=GEAR')
C
C
      IF ( IMODE .EQ. 1 ) THEN
        READ (ITAPE,*) NC
      ELSE
        READ *,NC
      ENDIF
```

```fortran
      WRITE (ITAPEW,*) NC,'   NEW COMMAND'
C
      IF (NC .EQ. 6) THEN
C
        IF (IGEAR .EQ. 0) THEN
          IGEAR=1
          PRINT 100
  100     FORMAT(1X,'GEAR DOWN')
C
        ELSE
          IGEAR=0
          PRINT 101
  101     FORMAT(1X,'GEAR UP')
C
        END IF
C
      ELSE IF (NC .EQ. 5) THEN
        IF ( IMODE .EQ. 0 ) THEN
          PRINT 81
   81     FORMAT(1X,'ILS OPTION (1=YES,0=NO)')
          READ *,ILS
        ELSE
          READ (ITAPE,*) ILS
        ENDIF
        WRITE (ITAPEW,*) ILS,'  ILS MODE SELECT, 1 = ON'
C
      ELSE IF (NC .EQ. 4) THEN
        IF ( IMODE .EQ. 0 ) THEN
          PRINT 71
   71     FORMAT(1X,'TRACE OPTION (1=YES)')
          READ *,ITRACE
        ENDIF
C
      ELSE IF (NC .EQ. 3) THEN
        IF ( IMODE .EQ. 1 ) THEN
          READ (ITAPE,*) HCOM
          READ (ITAPE,*) INAV
        ELSE
          PRINT 21
   21     FORMAT(1X,'NEW HEADING')
          READ *,HCOM
          PRINT 300
  300     FORMAT(1X,'1=NAV HEADING, 0=VECTOR')
          READ *,INAV
        ENDIF
        WRITE (ITAPEW,*) HCOM, '   NEW HEADING'
        WRITE (ITAPEW,*) INAV, '   0 = VECTOR, 1 = NAV'
C
        IF (INAV .EQ. 1) THEN
          PSIGC=HCOM
          CALL WIND(HP,VWO,VT,VGDUM,PSIGC,PSIW,HCOM,1)
C
        END IF
C
      ELSE IF (NC .EQ. 1) THEN
```

```
          IF (IMODE .EQ. 1 ) THEN
            READ (ITAPE,*) VCCOM
          ELSE
            PRINT 22
   22       FORMAT(1X,'NEW AIRSPEED OR MACH')
            READ *,VCCOM
          END IF
          WRITE (ITAPEW,*) VCCOM,'  NEW SPEED'
C
        ELSE IF (NC .EQ. 2) THEN
          IF (IMODE .EQ. 1 ) THEN
            READ (ITAPE,*) HPCOM
          ELSE
            PRINT 23
   23       FORMAT(1X,'NEW ALTITUDE')
            READ *,HPCOM
          ENDIF
          WRITE (ITAPEW,*) HPCOM,'  NEW ALTITUDE'
C
        ELSE IF (NC .EQ. 0) THEN
          IF ( IMODE .EQ. 1 ) THEN
            READ (ITAPE,*) SDIST
          ELSE
            PRINT 51
   51       FORMAT(/,'SEGMENT DISTANCE')
            READ *,SDIST
          END IF
          WRITE (ITAPEW,*) SDIST,'  SEGMENT DISTANCE'
          RETURN
C
        END IF
C
        GO TO 11
C
        END
        SUBROUTINE ATMOS(HP,VT,VC,XMN,Q,DEL)
        THETA=(1.-.00000687*HP)
        SIG=THETA**4.256
        DEL=THETA**5.256
        RHO=.002377*SIG
        Q=.5*RHO*((VT*1.688)**2)
        XMN=SQRT(Q/(1481.*DEL))
        QC=Q*(XMN*XMN/4.+1.)
        VC=1481.*SQRT((1.+QC/2116.)**.286-1.)
        RETURN
        END
        SUBROUTINE ENG(HP,XMN,ENHP,ENXMN,TTBL,TMAX,TMIN)
C
C ***********************************
C
C THIS ROUTINE CALCULATES :
C
C     TMAX (MAXIMUM CLIMB THRUST)
C
C     TMIN (IDLE THRUST)
```

```
C
C
C FOR GIVEN ALTITUDE AND MACH NUMBER
C
C ****************************************
C
      DIMENSION ENHP(8),ENXMN(8),TTBL(10,5,2)
      IHP=2
   10 IF(HP.LE.ENHP(IHP))GO TO 11
      IHP=IHP+1
      IF(IHP.GT.3)GO TO 11
      GO TO 10
   11 IXMN=2
   20 IF(XMN.LE.ENXMN(IXMN))GO TO 21
      IXMN=IXMN+1
      IF(IXMN.GT.4)GO TO 21
      GO TO 20
   21 SLXMN=(XMN-ENXMN(IXMN-1))/(ENXMN(IXMN)-ENXMN(IXMN-1))
      SLHP=(HP-ENHP(IHP-1))/(ENHP(IHP)-ENHP(IHP-1))
      TMAX1=SLXMN*(TTBL(IHP-1,IXMN,1)-TTBL(IHP-1,IXMN-1,1))
      TMIN1=SLXMN*(TTBL(IHP-1,IXMN,2)-TTBL(IHP-1,IXMN-1,2))
      TMAX2=SLXMN*(TTBL(IHP,IXMN,1)-TTBL(IHP,IXMN-1,1))
      TMIN2=SLXMN*(TTBL(IHP,IXMN,2)-TTBL(IHP,IXMN-1,2))
      TMAX1=TMAX1+TTBL(IHP-1,IXMN-1,1)
      TMIN1=TMIN1+TTBL(IHP-1,IXMN-1,2)
      TMAX2=TMAX2+TTBL(IHP,IXMN-1,1)
      TMIN2=TMIN2+TTBL(IHP,IXMN-1,2)
      TMAX=TMAX1+SLHP*(TMAX2-TMAX1)
      TMIN=TMIN1+SLHP*(TMIN2-TMIN1)
      RETURN
      END
      SUBROUTINE BANK(PSI,PSICOM,PHI)
C
C ****************************************
C
C THIS ROUTINE COMPUTES A BANK ANGLE
C TO CORRECT HEADING ERROR.
C
C  PSI = CURRENT HEADING OR COARSE
C
C  PSICOM = DESIRED HEADING OR COARSE
C
C PHI INCREMENTED 1 DEG PER ITERATION
C
C PHIMAX LIMITS BANK BASED ON PSIERR
C
C ****************************************
C
      DPHI=1.
      PSCOMT=PSICOM
      PSIT=PSI
      IF(PSI.GT.180.)PSIT=PSIT-360.
      IF(PSCOMT.GT.180.)PSCOMT=PSCOMT-360.
      PSIERR=PSCOMT-PSIT
      IF(ABS(PSIERR) .LT. .1 )GO TO 15
```

```fortran
      PHIMAX = 5.*ABS(PSIERR)
      IF ( PHIMAX .GT. 25. ) PHIMAX = 25.
      IF(PSIERR.LT.0.)GO TO 11
      IF(PSIERR.GT.180.)GO TO 10
      GO TO 20
   11 IF(PSIERR.LT.-180.)GO TO 20
      GO TO 10
   15 PSI=PSICOM
      PHI=0.
      GO TO 30
   10 PHI=PHI-DPHI
      IF(PHI.LT.-PHIMAX)PHI=-PHIMAX
      GO TO 30
   20 PHI=PHI+DPHI
      IF(PHI.GT.PHIMAX)PHI=PHIMAX
   30 RETURN
      END
      SUBROUTINE AERO(IFLAP,IGEAR,ISPOIL,ALPHA,CL,CD)
C
C ***************************************************
C
C THIS ROUTINE COMPUTES CL AND CD FOR
C SPECIFIED FLAP, GEAR AND SPOILER
C CONFIGURATION.
C
C   IFLAP = INDEX CORRESPONDING TO FLAP
C           SETTING IN TBFLAP ARRAY.
C
C   IGEAR  = 0 (GEAR UP), 1 (GEAR DN)
C
C   ISPOIL = 0 (SPOILERS RETRACTED), 1 (EXTENDED)
C
C ***************************************************
C
      COMMON/TB/TBCLA(8),TBCLAO(8),TBCLS(8),TBCDO(8),TBCDCL(8),
     *TBCDS(8),TBCDG(8),TBFLAP(8),XS(1000),YS(1000),
     *TIME(1000),VSAV(1000),HPSAV(1000),PSISAV(1000),
     *VCSAV(1000)
      CL=TBCLAO(IFLAP)+ALPHA*TBCLA(IFLAP)
      CD=TBCDO(IFLAP)+TBCDCL(IFLAP)*CL*CL
      IF(IGEAR.EQ.1)CD=CD+TBCDG(IFLAP)
      IF(ISPOIL.EQ.0)GO TO 20
      CL=CL+TBCLS(IFLAP)
      CD=CD+TBCDS(IFLAP)
   20 RETURN
      END
      SUBROUTINE FLAPS(FLAPL,FLAPH,VC,IFLAP)
C
C ***************************************************
C
C THIS ROUTINE DETERMINES PROPER FLAP
C SETTING BASED ON NOMINAL AIRSPEED
C SCHEDULE.
C
C ***************************************************
```

```
C
      DIMENSION FLAPL(8),FLAPH(8)
   10 VL=FLAPL(IFLAP)
      IF(VC.GT.VL)GO TO 11
      IFLAP=IFLAP+1
      IF(IFLAP.GT.8)GO TO 20
      GO TO 10
   11 VH=FLAPH(IFLAP)
      IF(VC.LT.VH)GO TO 12
      IFLAP=IFLAP-1
      IF(IFLAP.GE.1)GO TO 11
   20 IFLAP=0
   12 RETURN
      END
      SUBROUTINE SOLVE(WT,ALPHA,GAMMA,PSI,PHI,T,XL,
     *D,X,Y,HP,DR,ROC,VG,VT,PSIM)
C
C *****************************************
C
C THIS ROUTINE SOLVES THE TRAJECTORY
C EQUATIONS OF MOTION.
C
C *****************************************
C
   90 FORMAT(1X,'SOLVE ',4F10.3)
      DT=.2
      A=ALPHA/57.3
      G=GAMMA/57.3
      PS=PSI/57.3
      PH=PHI/57.3
      XM=WT/32.2
      VT=VT*1.688
      VG=VG*1.688
C
C CALCULATE "DOT" TERMS
C
      VDOT=(T*COS(A)-D-WT*SIN(G))/XM
      PSDOT=(XL+T*SIN(A))*SIN(PH)/(XM*VG*COS(G))
      GDOT=(XL+T*SIN(A))*COS(PH)/(XM*VG)-32.2*COS(G)/VG
      XDOT=VG*COS(G)*COS(PS)
      YDOT=VG*COS(G)*SIN(PS)
      HDOT=VG*SIN(G)
C
C INTEGRATE
C
      VT=(VT+VDOT*DT)/1.688
      VG=(VG+VDOT*DT)/1.688
      PSIM=PSIM+PSDOT*DT*57.3
      PSI=(PS+PSDOT*DT)*57.3
      GAMMA=(G+GDOT*DT)*57.3
      DX=XDOT*DT
      DY=YDOT*DT
      DR=SQRT(DX*DX+DY*DY)/6076.
      X=X+DX
      Y=Y+DY
```

```fortran
      HP=HP+HDOT*DT
C
C CORRECT HEADING
C
      IF(PSI.GT.360.)PSI=PSI-360.
      IF(PSI.LT.0.)PSI=PSI+360.
      IF(PSIM.GT.360.)PSIM=PSIM-360.
      IF(PSIM.LT.0.)PSIM=PSIM+360.
   91 FORMAT(1X,'SOLVE2',4F10.3)
      ROC=HDOT*60.
      RETURN
      END
      SUBROUTINE PLOT(ICNT)
C
C ***********************************************
C
C THIS ROUTINE USES PLOT10 FOR INTERACTIVE
C PLOTTING OF :
C
C   GROUND TRACK    (XLOC)
C
C   TIME HISTORIES
C
C     ALTITUDE      (HP)
C     SPEED         (V)
C       GROUNDSPEED (VG)
C       KCAS          (VC)
C
C ***********************************************
C
      COMMON/TB/TBCLA(8),TBCLAO(8),TBCLS(8),TBCDO(8),TBCDCL(8),
     *TBCDS(8),TBCDG(8),TBFLAP(8),XS(1000),YS(1000),
     *TIME(1000),VSAV(1000),HPSAV(1000),PSISAV(1000),
     *VCSAV(1000)
      DIMENSION X(200),Y(200)
    1 PRINT 100
  100 FORMAT(1X,'X VARIABLE',/,1X,'1=XLOC,2=TIME')
      READ *,IX
      IF(IX.EQ.1)GO TO 11
      PRINT 101
  101 FORMAT(1X,'Y VARIABLE',/,1X,'1=V,2=ALT')
      READ *,IY
   11 CALL INITT(120)
      CALL BINITT
      CALL SLIMX(300,900)
      CALL XFRM(3)
      CALL XMFRM(3)
      CALL YFRM(3)
      CALL YMFRM(3)
      IF(IX.GT.1)GO TO 10
      XS(1)=ICNT-1
      YS(1)=ICNT-1
      CALL ZLINE(0)
      PRINT 103
  103 FORMAT(1X,'RANGE SCALE (NM)')
```

```
        READ *,SCALE
        XLOW=-SCALE
        XHIGH=SCALE
        YLOW=-SCALE
        YHIGH=SCALE
        PRINT 104                .
    104 FORMAT(1X,'TIME INTERVAL FOR TIC MARKS')
        READ *,ITIC
        ITIC=ITIC/2
        CALL DLIMX(XLOW,XHIGH)
        CALL DLIMY(YLOW,YHIGH)
        CALL CHECK(YS,XS)
        CALL DSPLAY(YS,XS)
        IT=1
        ITT=ITIC
        DO 60 I=2,ICNT
        ITT=ITT+1
        IF(ITT.LT.ITIC)GO TO 60
        IT=IT+1
        X(IT)=XS(I)
        Y(IT)=YS(I)
        ITT=0
     60 CONTINUE
        X(1)=IT-1
        Y(1)=IT-1
        CALL ZLINE(-1)
        CALL SYMBL(8)
        CALL CHECK(Y,X)
        CALL CPLOT(Y,X)
C
C       ** VOR SITES **
C
C DEN, IOC, GUC
C
        X(1)=3.
        Y(1)=3.
        X(2)=-1.79
        Y(2)=2.40
        X(3)=27.69
        Y(3)=-13.59
        X(4)=-82.10
        Y(4)=-99.13
        CALL SYMBL(1)
        CALL CHECK(X,Y)
        CALL CPLOT(X,Y)
C
C       ** WAYPOINTS **
C
C ENROUTE:
C
C  ELBEC, ACREE, SHREW, BENAM, SMITY, RAMAH
C
        X(1)=15.
        Y(1)=15.
        X(2)=-61.24
```

```
            Y(2)=-78.99
            X(3)=-42.54
            Y(3)=-60.93
            X(4)=-30.62
            Y(4)=-42.34
            X(5)=-36.14
            Y(5)=34.47
            X(6)=39.97
            Y(6)=42.97
            X(7)=45.91
            Y(7)=-26.68
C
C TERMINAL:
C
C  BYSON, DRAKO, KEANN, TROZE, JASIN, FLOTS, WIFES
C  ALTUR, WATKI
C
            X(8)=-21.43
            Y(8)=-28.09
            X(9)=-25.74
            Y(9)=24.74
            X(10)=22.54
            Y(10)=26.04
            X(11)=-9.71
            Y(11)=-9.99
            X(12)=-12.74
            Y(12)=12.64
            X(13)=10.26
            Y(13)=14.10
            X(14)=13.14
            Y(14)=-5.70
            X(15)=5.35
            Y(15)=1.16
            X(16)=13.47
            Y(16)=3.16
            CALL SYMBL(3)
            CALL CHECK(X,Y)
            CALL CPLOT(X,Y)
            CALL ZLINE(1)
            CALL SYMBL(0)
C
C        ** RANGE RINGS **
C
            IF(SCALE .GT. 40.) GO TO 53
            X(1)=180.
            Y(1)=180.
            R=0.
            DO 50 I=1,3
            R=R+10.
            TH=0.
            DO 52 J=2,181
            TH=TH+.0349
            X(J)=R*COS(TH)
            Y(J)=R*SIN(TH)
         52 CONTINUE
```

```
      CALL CHECK(X,Y)
      CALL CPLOT(X,Y)
   50 CONTINUE
   53 CONTINUE
C
C     ** EXTENDED CENTER-LINE **
C
      X(1)=2.
      Y(1)=2.
      X(2)=0.
      Y(2)=0.
      X(3)=19.5
      Y(3)=4.5
      CALL CHECK(X,Y)
      CALL CPLOT(X,Y)
      CALL MOVABS(350,700)
      CALL TSEND
      ITIC=2*ITIC
      PRINT 105,ITIC
  105 FORMAT(1X,'TIME INTERVAL=',I5,' SEC')
      GO TO 900
   10 TIME(1)=ICNT-1
      IF(IY.GT.1)GO TO 20
      PRINT 250
  250 FORMAT(1X,'1=GNDSPEED,0=CAS')
      READ *,ISP
      PRINT 251
  251 FORMAT(1X,'YMIN,YMAX,')
      READ *,YMIN,YMAX
      CALL DLIMY(YMIN,YMAX)
      IF(ISP.EQ.1)GO TO 12
      VCSAV(1)=ICNT-1
      CALL CHECK(TIME,VCSAV)
      CALL DSPLAY(TIME,VCSAV)
      GO TO 900
   12 VSAV(1)=ICNT-1
      CALL CHECK(TIME,VSAV)
      CALL DSPLAY(TIME,VSAV)
      GO TO 900
   20 HPSAV(1)=ICNT-1
      PRINT 252
  252 FORMAT(1X,'YMIN,YMAX,')
      READ *,YMIN,YMAX
      CALL DLIMY(YMIN,YMAX)
      CALL CHECK(TIME,HPSAV)
      CALL DSPLAY(TIME,HPSAV)
  900 CONTINUE
      CALL MOVABS(0,700)
      CALL TSEND
      PRINT 102
  102 FORMAT(1X,'NEW PLOT (1=YES)')
      READ *,IC
      IF(IC.EQ.1)GO TO 1
      RETURN
      END
```

34

```
          SUBROUTINE ILS26L(VT,VG,X,Y,HP,HPCOM,PSI,PSIW,PSICOM,ILOC,
     *IGLIDE,VWO,PSIM)
C
C ***********************************************
C
C THIS ROUTINE HANDLES ILS APPROACH TO
C RUNWAY 26L AT DENVER.
C
C COMMANDED HEADING (PSICOM) MODIFIED
C TO CAPTURE RUNWAY EXTENDED CENTERLINE.
C
C COMMANDED ALTITUDE (HPCOM) MODIFIED
C TO FLY GLIDESLOPE.
C
C ***********************************************
C
          PSIRNY=257.
          XTHRS=0.
          YTHRS=0.
          DXTHRS=XTHRS-Y/6076.
          DXTHRS=ABS(DXTHRS)
          DYLOC=ABS(DXTHRS*.231+YTHRS-X/6076.)
          IF(ILOC.EQ.1)GO TO 10
          DELPSI=ABS(PSI-PSIRNY)
          IF(DELPSI.GT.180.)DELPSI=DELPSI-180.
          PSIDOT=57.3*32.2*TAN(25./57.3)/(VT*1.688)
          TIME=DELPSI/(PSIDOT*3600.)
          DYTURN=SIN(DELPSI/114.6)*VT*TIME
C         PRINT 110,DYTURN,DYLOC
  110 FORMAT(1X,'DYTURN=',F10.4,' DYLOC=',F10.4)
          IF(DYLOC.GT.DYTURN)GO TO 20
          ILOC=1
   10 CONTINUE
          IF ( VWO .NE. 0. ) GO TO 11
          PSICOM = PSIRNY
          GO TO 12
   11 CALL WIND(HP,VWO,VT,VG,PSIRNY,PSIW,PSICOM,1)
   12 IF (DYLOC .GT. .017) GO TO 20
          PSI = PSIRNY
          IF ( VWO .EQ. 0. ) PSIM = PSI
C
C        ** GLIDESLOPE CHECK **
C
   20 XTD=ABS(DXTHRS)+1000./6076.
          HPGSLP=5330.+.0523*XTD*6076.
          DHP=HPGSLP-HP
          IF(ABS(DHP).LT.50.)GO TO 30
          IGLIDE=0
          IF(DHP.LT.0.)HPCOM=HPGSLP
          GO TO 40
   30 IGLIDE=1
          HPCOM=HP
   40 IF(HP.LE.5330.)HPCOM=5330.
          RETURN
          END
```

```
      SUBROUTINE WIND(HP,VWO,VT,VG,PSI,PSIW,PSIM,INAV)
C
C *******************************************************
C
C THIS ROUTINE CORRECTS GROUNDSPEED AND HEADING
C FOR WIND EFFECTS.
C
C    PSI  = GROUND COURSE
C    PSIM = MAGNETIC HEADING
C
C INAV = 0 SIGNIFIES HEADING MODE :
C
C    GROUND COURSE COMPUTED BASED ON REFERENCE
C    MAG HEADING.
C
C INAV = 1 SIGNIFIES NAVIGATION MODE :
C
C    MAGNETIC HEADING COMPUTED BASED ON REFERENCE
C    GROUND COURSE.
C
C *******************************************************
C
      IF(VWO.EQ.0.)GO TO 40
      VW=VWO+(HP-5330.)*.0012
      PSIWV=PSIW-180.
      IF(PSIWV.LT.0.)PSIWV=PSIWV+360.
      IF(INAV.EQ.1)GO TO 20
   10 DPSIW=(PSIWV-PSIM)/57.3
      DPSI=ATAN(VW*SIN(DPSIW)/(VT+VW*COS(DPSIW)))
      PSI=PSIM+DPSI*57.3
      GO TO 30
   20 DPSI=ASIN(VW*SIN((PSIWV-PSI)/57.3)/VT)
      PSIM=PSI-DPSI*57.3
      DPSIW=(PSIWV-PSIM)/57.3
   30 VG=VT*COS(DPSI)+VW*COS(DPSIW-DPSI)
      RETURN
   40 VG=VT
      PSI=PSIM
      RETURN
      END
--EOR--
--EOI/TOP--
??
```

REFERENCES

1. Williams, David H.: Time-Based Self-Spacing Technique Using Cockpit Display of Traffic Information During Approach to Landing in a Terminal Area Vectoring Environment. NASA TM-84601, 1983.

2. Williams, David H. and Moen, Gene C.: Simulation Study of Traffic-Sensor Noise Effects on Utilization of Traffic Situation Display for Self-Spacing Task. NASA TP-2082, AVRADCOM TR 82-B-8, 1983.

3. Department of Transportation, Federal Aviation Administration: Air Traffic Control Handbook 7110.65C (change 6). November 24, 1983.

4. Department of Transportation, Federal Aviation Administration: Airman's Information Manual (AIM) Basic Flight Information and ATC Procedures. November 24, 1983.

5. Praft and Whitney Aircraft: JT8D Commercial Turbofan Engine Installation Handbook. June 1, 1975 revision.

Table I.- Aircraft in Sample Scenario

Aircraft	Starting Position	Ending Position	Time to Merge Point Desired	Actual	Time Bias
Reference	ELBEC	Runway 26L	1330	1330	0
1	SMITY	Runway 26L	890	710	180
2	RAMAH	Runway 26L	1030	670	360
3	BACCA	Runway 26L	1130	1540	-410
4	ELBEC	Runway 26L	1230	1330	-100
5	SMITY	Runway 26L	1530	710	820
6	BENAM	Runway 26L	1630	890	740
7	ELBEC	Runway 26L	1710	1300	410
8	Runway 35L	Out Bound South	500	240	260
9	Runway 35L	Out Bound South	750	240	510
10	Runway 35L	Out Bound South	920	240	680
11	Runway 35L	Out Bound East	1230	200	1030

NOTE: Merge point for aircraft ending at runway 26L was the outer marker.

Merge point for the departures was the horizontal position where their ground track crossed the ground track of the reference aircraft.

All times expressed in seconds.

ARTCC BOUNDARY
SECTOR BOUNDARY
SIMULATION AREA
VORTAC STATION

100 N. MI.

0

DENVER ARTCC
HIGH ALTITUDE SECTORS

DEN

DENVER ARTCC
LOW ALTITUDE SECTORS

DEN
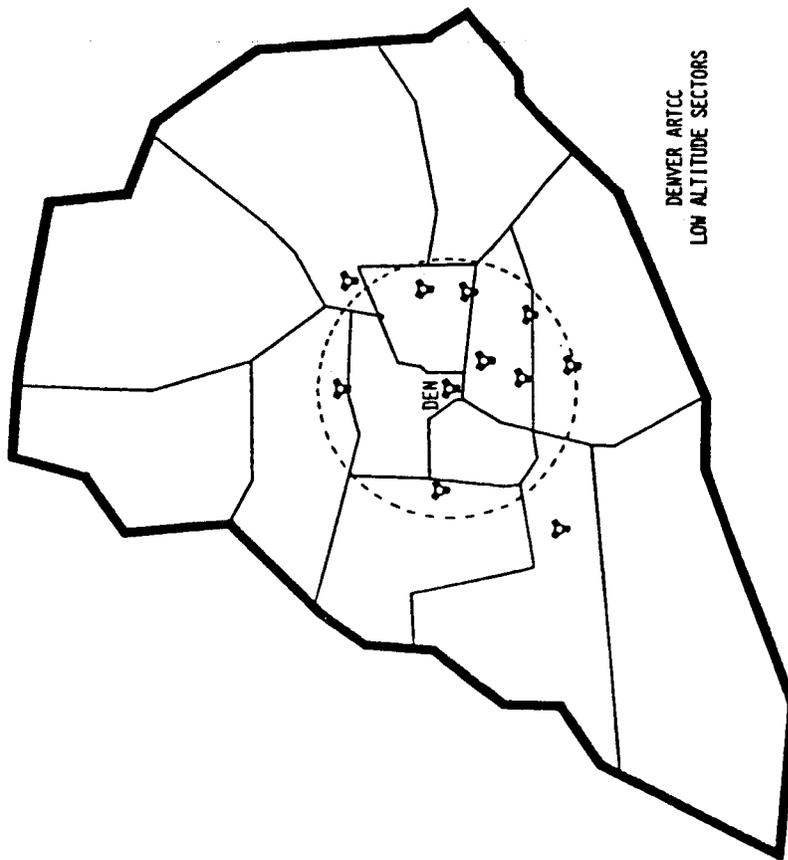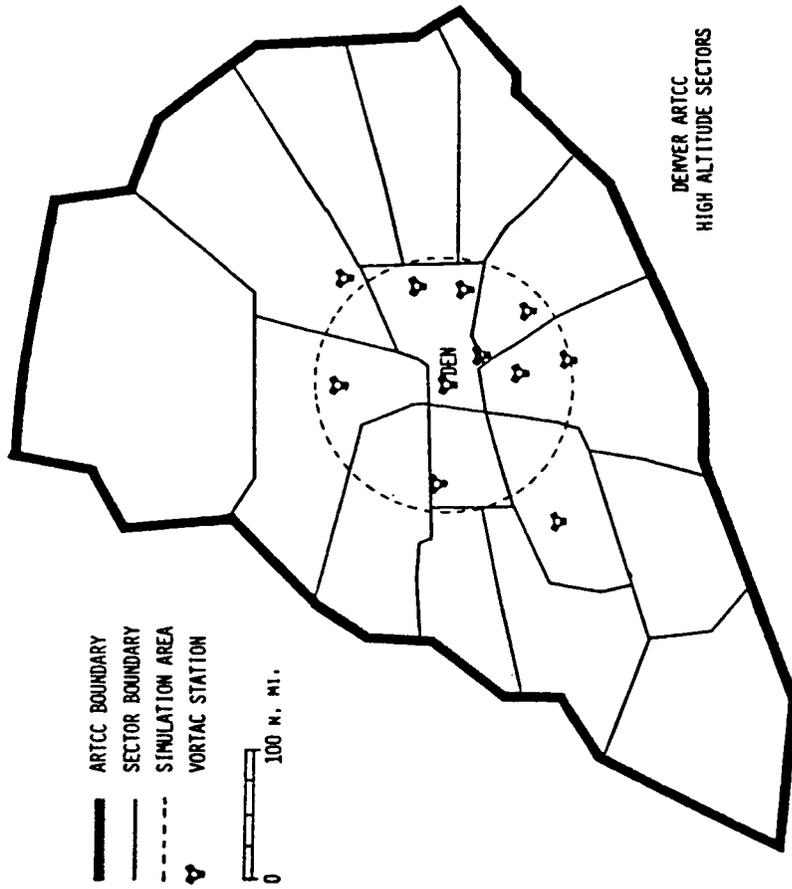
Figure 1.- Horizontal view of Denver ARTCC showing area
modeled in traffic scenario.

DRAKO △

△ KEANN

DEN

KIOWA

BYSON △

0   20 n.mi.

DEPARTURE/
SATELLITE

APPROACH
CORRIDORS

FINAL APPROACH
SEQUENCING FOR
RUNWAY 26L
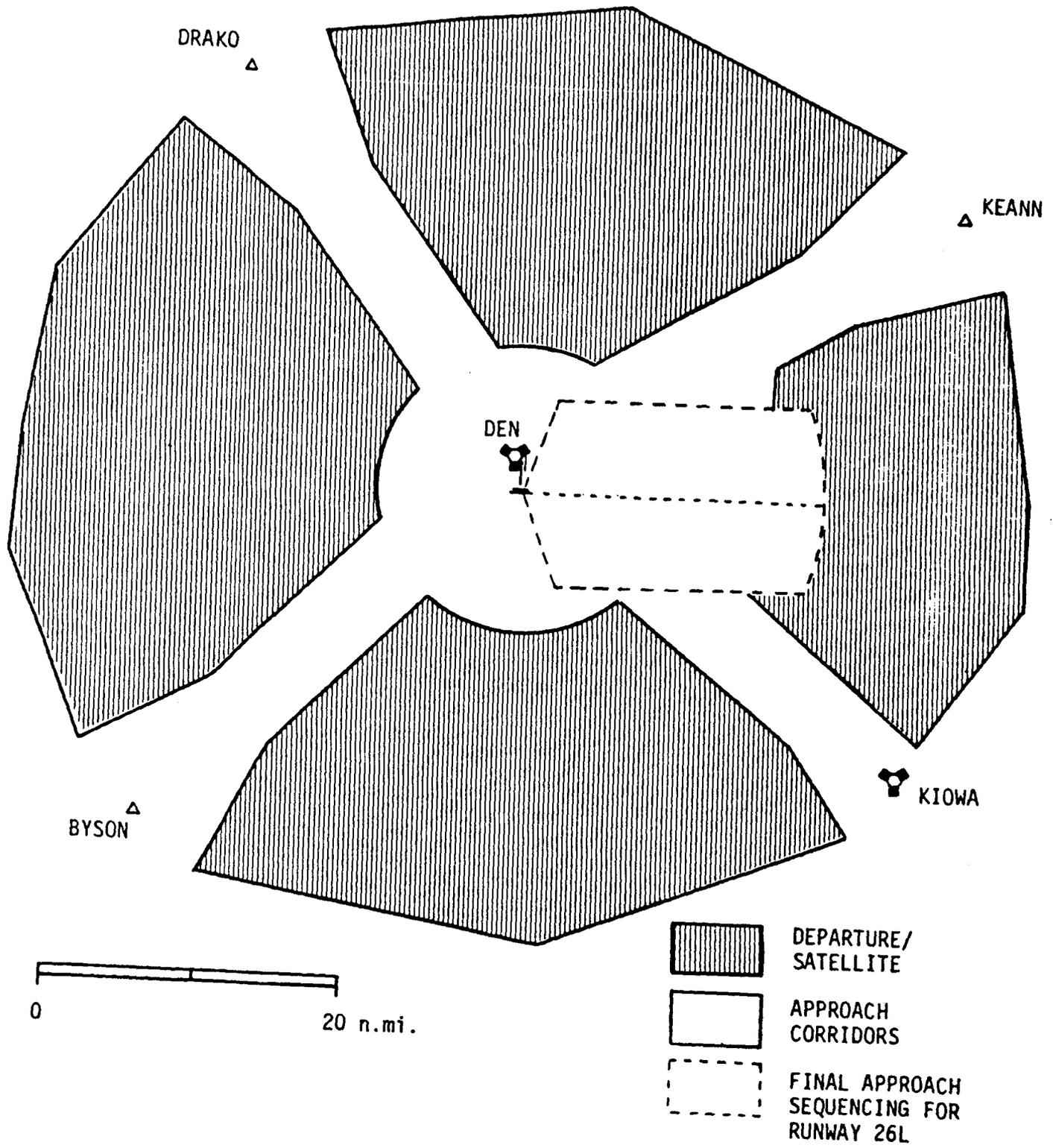
Figure 2.- Horizontal view of Denver Terminal Radar
Approach Control (TRACON) airspace.

x 10³

Altitude above MSL, ft

30

20

10

0

I

II

III

II

IV*

V

100
West

0

100
East

Range, n. mi.
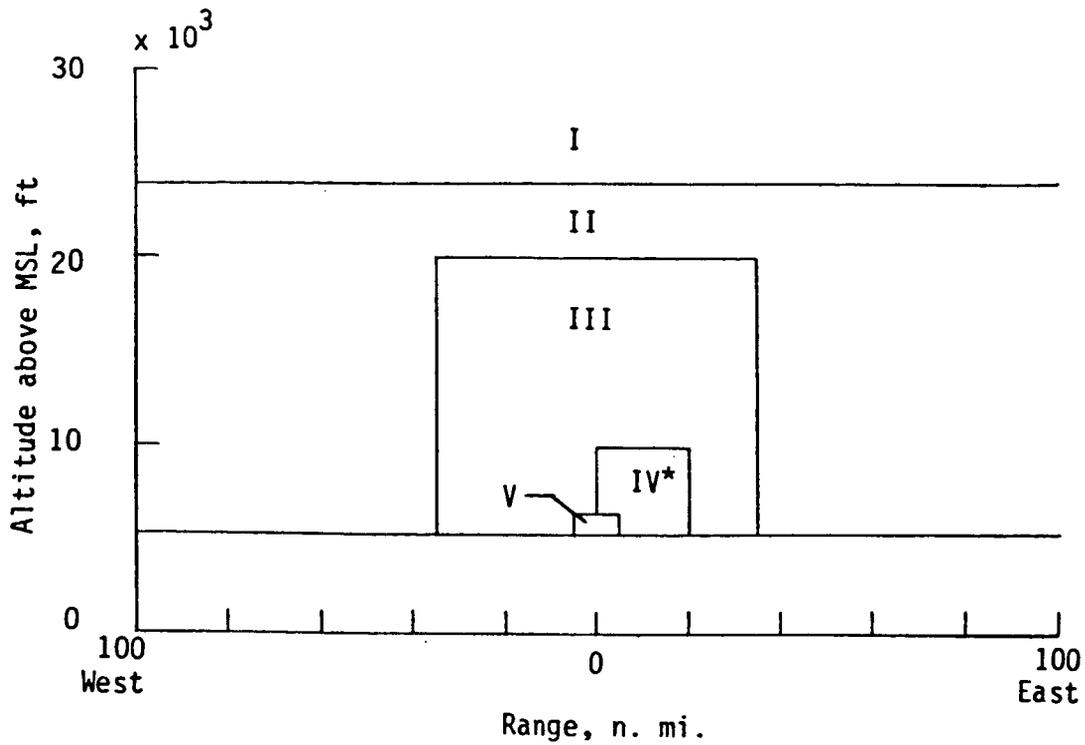
Airspace segment	Altitude; Typical distance from airport; Responsible controller(s)	Significant events
I	Above 24 000 ft; Greater than 80 n.mi.; High altitude enroute	Enroute cruise, descent clearance
II	20 000 to 24 000 ft; Greater than 35 n.mi.; Low altitude enroute	Profile descent, enroute climb clearance
III	Surface to 20 000 ft; 10 to 35 n.mi.; Approach and departure sectors	Vectored to final approach zone or departure course
IV*	7 000 to 10 000 ft; 5 to 20 n.mi. east; Final approach control	Vectored to ILS
V	Surface to 6 500 ft; 0 to 5 n.mi.; Local control	Landing/Takeoff clearance

*Note: Runway 26L arrival configuration

Figure 3.- Vertical view of simulated airspace.
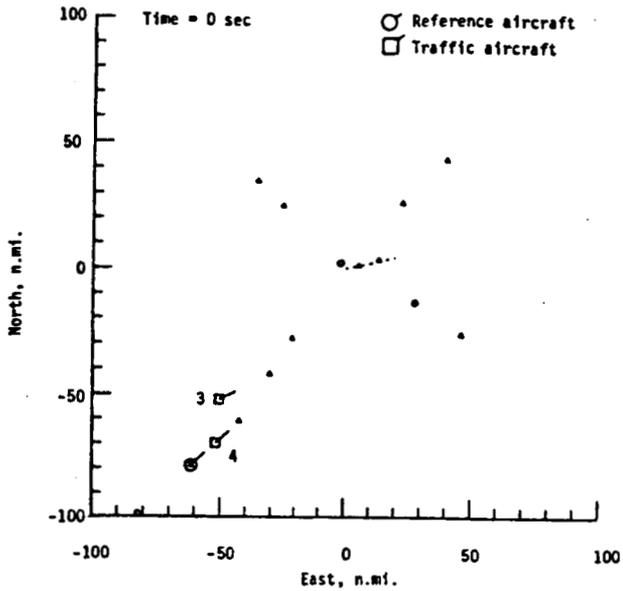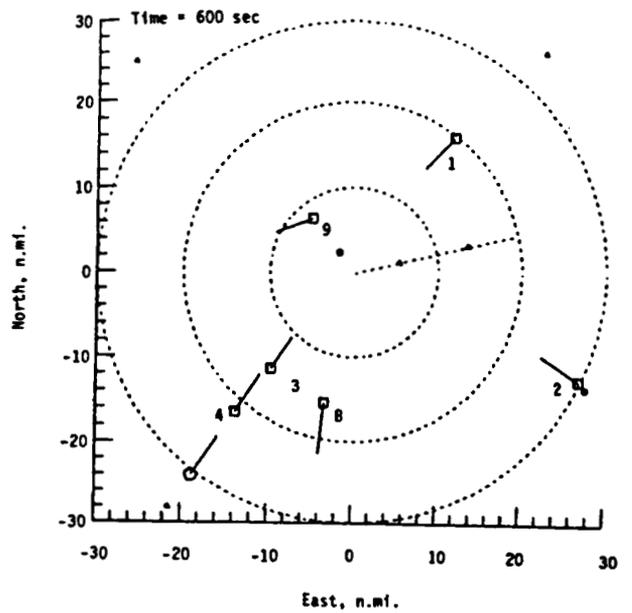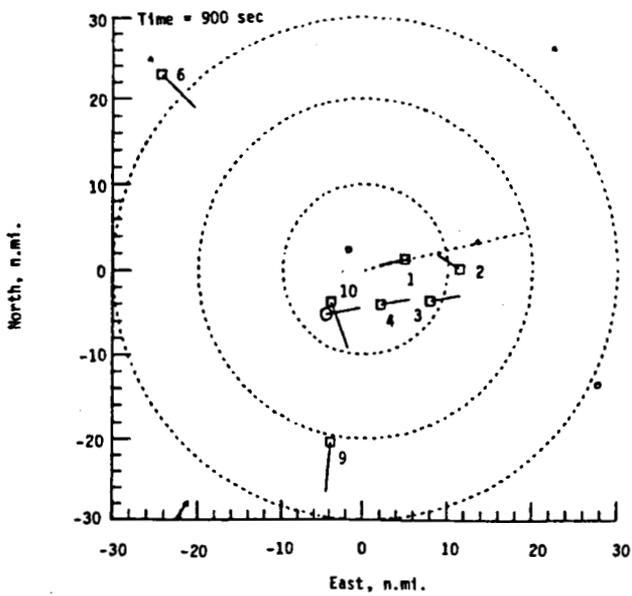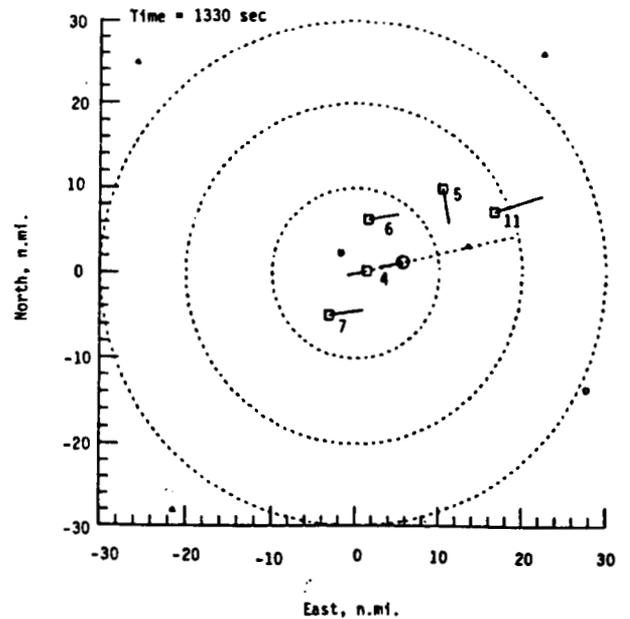
Figure 4.- Horizontal traffic situation at various times during
the sample scenario.

Legend:
△ Navigation fix
○ VORTAC station
+ Time mark (100 sec interval)
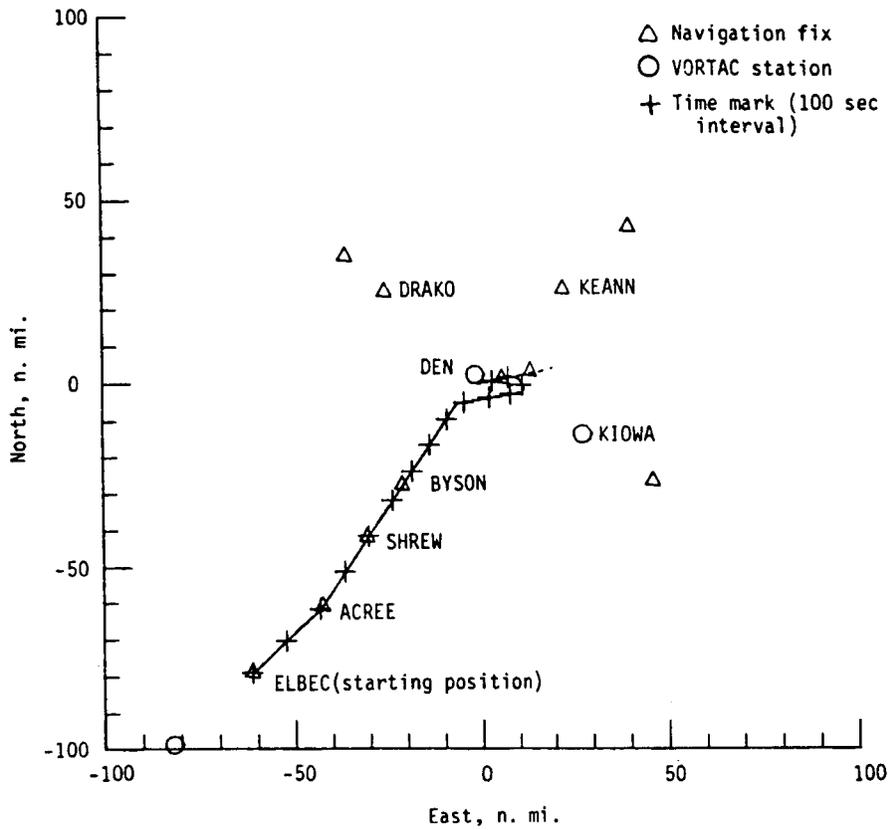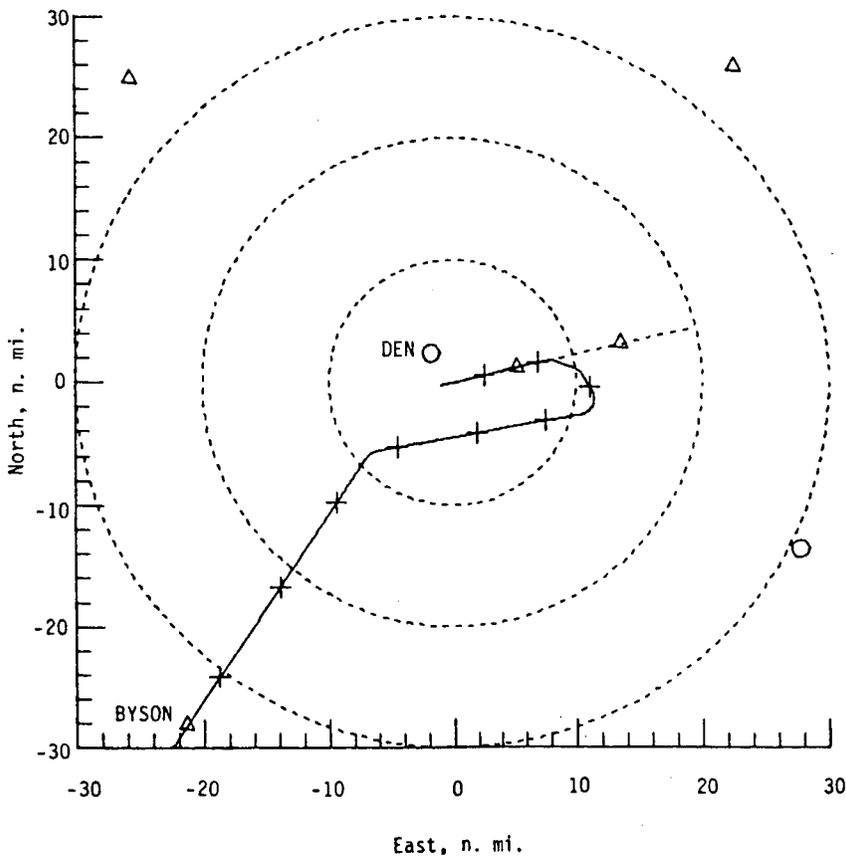
(a) 100 n. mi. range.

(b) 30 n. mi. range.

Figure A1.- Horizontal ground track of sample profile.
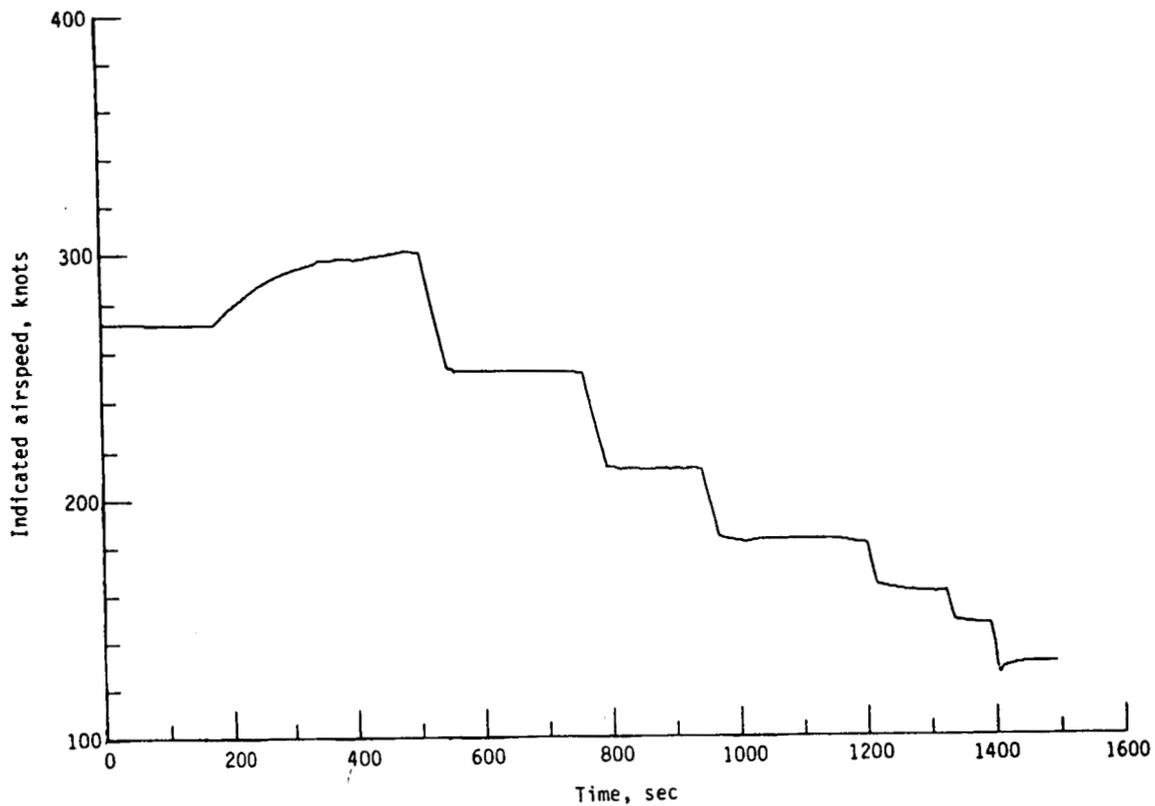
(b) Airspeed profile.
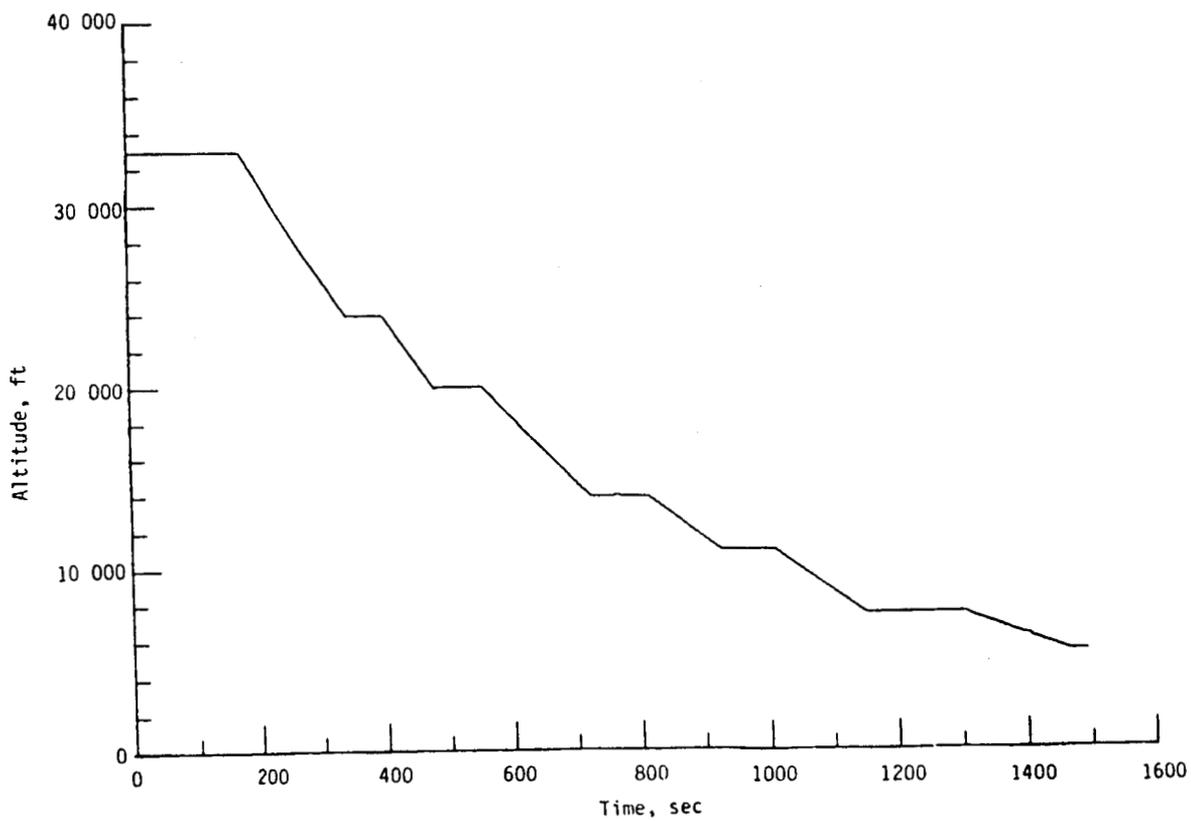
(a) Altitude profile.

Figure A2.- Time histories of sample profile.

1. Report No. NASA TM-86397	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Traffic Scenario Generation Technique for Piloted Simulation Studies		5. Report Date April 1985
		6. Performing Organization Code 505-35-13-03
7. Author(s) David H. Williams Douglas C. Wells		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code

15. Supplementary Notes

David H. Williams: NASA Langley Research Center, Hampton, Virginia.
Douglas C. Wells: Embry-Riddle Aeronautical University, Daytona Beach, Florida.

16. Abstract

Piloted simulation studies of cockpit traffic display concepts require the development of representative traffic scenarios. With the exception of specific aircraft interaction issues, most research questions can be addressed using traffic scenarios consisting of prerecorded aircraft movements merged together to form a desired traffic pattern. Prerecorded traffic scenarios have distinct research advantages, allowing control of traffic encounters with repeatability of scenarios between different test subjects. This paper describes a technique for generation of prerecorded jet-transport traffic scenarios suitable for use in piloted simulation studies. Individual flight profiles for the aircraft in the scenario are created interactively with a computer program designed specifically for this purpose. The profiles are then time-correlated and merged into a complete scenario. This technique has been used to create traffic scenarios for the Denver, Colorado area with operations centered at Stapleton International Airport. Traffic scenarios for other areas can also be created using this technique, with appropriate modifications made to the navigation fix locations contained in the flight profile generation program.

17. Key Words (Suggested by Author(s)) cockpit traffic display piloted simulation air traffic scenarios air traffic control	18. Distribution Statement Subject Category 03
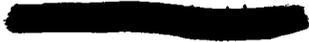
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 45	22. Price